



# REALIZACE PROGRAMU PRO OPTIMALIZACE TVARU A VELIKOSTI VNITŘNÍCH PÁNEVNÍCH DLAH Z ŘEZŮ CT

## Bakalářská práce

*Studijní program:* B3944 – Biomedicínská technika  
*Studijní obor:* 3901R032 – Biomedicínská technika  
*Autor práce:* **Bc. Martin Huněk**  
*Vedoucí práce:* MUDr. Jaroslav Šrám





# IMPLEMENTATION OF PROGRAM FOR SHAPE AND DIMENSIONS OPTIMIZATION OF PELVIS INTERNAL FIXATOR BASED UPON COMPUTER-ASSISTED TOMOGRAPHY IMAGES

**Bachelor thesis**

*Study programme:* B3944 – Biomedical technology  
*Study branch:* 3901R032 – Biomedical technology  
*Author:* **Bc. Martin Huněk**  
*Supervisor:* MUDr. Jaroslav Šrám



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin Huněk**  
Osobní číslo: **Z12000007**  
Studijní program: **B3944 Biomedicínská technika**  
Studijní obor: **Biomedicínská technika**  
Název tématu: **Realizace programu pro optimalizace tvaru a velikosti  
vnitřních pánevních dlah z řezů CT**  
Zadávací katedra: **Ústav zdravotnických studií**

### Z á s a d y p r o v y p r a c o v á n í :

Cíle práce:

Vytvoření programu pro zajištění prostorových dispozic dlah z CT řezů.

Zpětná korelace sejmutých 2D řezů pánve s rozměry předpokládané dlahy.

Tvorba dokumentace k vytvořenému SW.

Teoretická východiska (včetně výstupu z BP):

Při práci se bude vycházet hlavně z dosud známých algoritmů pro analýzu obrazových dat v technické diagnostice. Důraz bude kladen hlavně na snadnost obsluhy lékařským personálem a minimalizace možností chyb lidského faktoru.

Výzkumné otázky:

Rozvaha nad volbou algoritmů.

Metoda:

Aplikovaný výzkum a experimentální vývoj.

Technika práce, vyhodnocení dat:

Empirické zhodnocení výstupních dat na základě grafické podobnosti.

Místo a čas realizace výzkumu:

Spínální jednotka, Krajská nemocnice Liberec, a.s.

Ústav mechatroniky a technické informatiky, Technická univerzita v Liberci.

Akademický rok 2013/2014.

Vzorek:

Iniciální vzorek minimálně 18 řezů. Dále dle potřeby navazujícího výzkumu realizovaného na spínální jednotce KNL a.s .

Rozsah grafických prací:

Rozsah pracovní zprávy:

**50 - 70 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**viz příloha**


Vedoucí bakalářské práce:

**MUDr. Jaroslav Šrám**


Ústav zdravotnických studií

Datum zadání bakalářské práce: **12. září 2013**

Termín odevzdání bakalářské práce: **30. dubna 2014**

  
prof. Dr. Ing. Zdeněk Kůs  
rektor



  
Mgr. Marie Froňková  
pověřena vedením ústavu

V Liberci dne 28. února 2014

# Příloha zadání bakalářské práce

## Seznam odborné literatury:

- ŠRÁM, J., S. TALLER, R. LUKÁŠ a L. ENDRYCH. Užití Omega dlahy při stabilizaci zlomenin acetabula - první zkušenosti. ACTA CHIRURGIAE ORTHOPAEDICAE ET TRAUMATOLOGIAE ČECHOSL. 2013, č. 80, 118 - 124. Dostupné z: <http://www.achot.cz/detail.php?stat=617>
- TALLER, S., ŠRÁM, J., LUKÁŠ, R., KŘIVOHLÁVEK, M.: Zlomeniny pánevního kruhu a acetabula operované přístupem dle Stoppy. Acta Chir. orthop. Traum. Čech., 77: 93-98, 2010.
- HLAVÁČ, Václav. Zpracování signálů a obrazů. 1. vyd. Praha: Vydavatelství ČVUT, 2002, 220 s. ISBN 80-010-2114-9.
- SEUL, Michael. Practical algorithms for image analysis: description, examples and code. Vyd. 1. New York: Cambridge University Press, 2005, 295 s. ISBN 05-216-6065-3.
- RUSS, John C, J RUSS. Introduction to image processing and analysis. Boca Raton: CRC Press, 2008, 355 s. ISBN 08-493-7073-6.
- BRADSKI, Gary R. Learning OpenCV. Sebastopol: O'Reilly, c2008, xvii, 555 s. ISBN 978-0-596-51613-0.
- NATIONAL INSTRUMENTS. LabView: Getting Started with LabVIEW [online]. Austin: National Instruments, 2012, 2012-06 [cit. 2013-11-20]. Dostupné z: <http://www.ni.com/pdf/manuals/373427h.pdf>
- WOHLER, Christian. 3d computer vision: efficient methods and applications. 2nd ed. London: Springer, 2013, 381 s. ISBN 978-1-4471-4149-5.
- Šonka M., Hlaváč V., Boyle R.: Image Processing, Analysis, and Machine Vision, PWS Publishing, 1998, ISBN 0-534-953-93.
- JAKSCH, Ivan. Základy digitálního obrazu: Vektorová a rastrová grafika. In: Přednášky z předmětu Zobrazovací systémy [online]. Liberec: Technická univerzita v Liberci, 2012 [cit. 2013-11-20]. Dostupné z: [http://www.fm.tul.cz/esf0247/download\\_file.php/ESF\\_zakl\\_digit\\_obr.pdf?id=838](http://www.fm.tul.cz/esf0247/download_file.php/ESF_zakl_digit_obr.pdf?id=838)
- CHALOUPKA, Josef. Filtrace šumu, hledání hran. In: Přednášky z předmětu: Snímání, digitalizace a zpracování obrazových dat [online]. Liberec: Technická univerzita v Liberci, 2013 [cit. 2013-11-20]. Dostupné z: <https://www.ite.tul.cz/vyuka/mod/resource/view.php?id=831>

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 26.4.2014

Podpis: 

## Abstrakt

Tato práce se zabývá návrhem postupů a tvorbou konkrétní implementace vyhodnocování prostorových dispozic pánve. Jako zdroj původních dat je v této práci uvažován předem definovaný CT řez předem vyexportovaný z formátu DICOM do formátu JPEG. Prvním výstupem této práce má být křivka rozhraní kost-ostatní tkáň z dodaného řezu (v pánevní rovině) v místech pro přikládání vnitřní dlahy. Druhým pak nástroj na porovnání pravé a levé strany pánve pro účely pozdějšího statistického zpracování. Posledním výstupem této práce bude dokumentace k vytvořeným programům včetně všech zdrojových kódů. Dosažené výsledky budou průběžně kontrolovány a konzultovány s praxí na pracovišti vedoucího práce.

Klíčová slova: počítačové vidění, úprava obrazu, grafické transformace, metoda nejmenších čtverců, LabView

## Abstract

This thesis describes the design of procedures and the creation of a specific implementation for evaluation of the spatial disposition of the pelvis. As the source of the original data in this thesis are considered a pre-defined CT slice exported from DICOM format to JPEG. The first output of this thesis is the curve of the border bone-other tissue from the supplied slice (pelvic plane) in place for attaching the inner plate. The second output is a tool to compare the right and left sides of the pelvis for later statistical processing. The final output of this thesis is documentation for the programs, including all the source code. The obtained results have been continuously reviewed and discussed with the practice at the workplace of the supervisor.

Keywords: computer vision, image editing, graphical transformations, least squares method, LabView

## Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce za vstřícný a trpělivý přístup a za rady ohledně realizace. Rovněž bych touto cestou chtěl poděkovat i dalším lidem, kteří mi poskytli cenné rady, či jakkoliv přispěli k realizaci této práce.

Jmenovitě bych chtěl poděkovat vedoucímu práce MUDr. Šrámovi (za obětavý a trpělivý přístup), konzultantovi Ing. Hančilovi (za pomoc při tvorbě programu), Mgr. Stebelovi, Ph.D. (za pomoc s volbou algoritmu prokládání kružnice křivkou) a vneposlední řadě doc. RNDr. Satrapovy, Ph.D. za tvorbu šablony pro tvorbu závěrečných prací v  $\text{\LaTeX}$ u pomocí které je tato práce vysázena.



# Obsah

Seznam zkratek . . . . .	12
<b>1 Úvod</b>	<b>13</b>
<b>2 K čemu to všechno je?</b>	<b>14</b>
<b>3 Vyhodnocování obrazových dat</b>	<b>16</b>
3.1 Prahování . . . . .	17
3.2 Hranová detekce . . . . .	18
3.3 Výběr křivky . . . . .	22
3.4 Seznámení s prostředím LabView . . . . .	23
3.5 Implementace programu . . . . .	25
3.5.1 První krok – načtení obrazu . . . . .	25
3.5.2 Druhý krok – filtrování obrazu . . . . .	27
3.5.3 Třetí krok – odblokování extrakce . . . . .	29
3.5.4 Čtvrtý krok – extrakce křivky . . . . .	29
3.5.5 Pátý a šestý krok – předzpracování a uložení . . . . .	31
3.6 Uživatelské rozhraní . . . . .	34
<b>4 Analýza získaných křivek</b>	<b>38</b>
4.1 Formát vstupních dat . . . . .	38
4.2 Grafické transformace . . . . .	39
4.2.1 Translace . . . . .	39
4.2.2 Rotace . . . . .	40
4.3 Analýza délky křivky . . . . .	41
4.4 Analýza křivosti . . . . .	43
4.4.1 Původní záměr . . . . .	43
4.4.2 Minimalizace funkce metodou nejmenších čtverců . . . . .	45
4.5 Analýza vzdálenosti křivek . . . . .	48
4.6 Implementace programu . . . . .	49
4.6.1 Funkce Circular fit . . . . .	50
4.6.2 Hlavní program – příprava a zobrazení dat . . . . .	50
4.6.3 Měření vzdálenosti křivek . . . . .	52
4.6.4 Měření délek . . . . .	55
4.6.5 Proložení kružnice a uložení výsledků . . . . .	55
4.7 Uživatelské rozhraní . . . . .	59

4.8 Výstupní data ke statistickému zhodnocení . . . . .	59
<b>5 Závěr</b>	<b>61</b>
<b>Přílohy</b>	<b>65</b>
<b>A Obsah CD</b>	<b>65</b>
<b>B Dokumentace k programu pro extrakci křivky</b>	<b>66</b>
<b>C Dokumentace k programu pro srovnání získaných křivek</b>	<b>83</b>

## Seznam obrázků

2.1	Omega dlaha (obrázek poskytnut vedoucím práce) . . . . .	15
3.1	Rozdíly v SI skloubení mezi pacienty . . . . .	16
3.2	Původní řez bez úprav . . . . .	17
3.3	Histogram s nastavením prahového filtru . . . . .	18
3.4	Řez po aplikování prahového filtru . . . . .	19
3.5	Řez po aplikování Laplaceova filtru . . . . .	20
3.6	Ukázka kódu v LabView (změna cesty k souboru) . . . . .	24
3.7	Načtení obrazu a zobrazení náhledu . . . . .	26
3.8	Prahování obrazu a filtrování částic . . . . .	28
3.9	Extrakce křivky . . . . .	30
3.10	Otočení a uložení křivky . . . . .	32
3.11	Náhled a nastavení prahování . . . . .	35
3.12	Příklad nastavení filtru částic . . . . .	36
3.13	Výběr oblasti zájmu . . . . .	37
4.1	Společná část kódování ASCII, CP1250 a UTF-8 (zdroj:[1]) . . . . .	38
4.2	Ukázka formátování dat v souboru křivky (zvýrazněn jeden řádek) . . . . .	39
4.3	Algoritmus pro výpočet délky křivky . . . . .	42
4.4	Nalezení středu úsečky pomocí kružítka . . . . .	44
4.5	Vliv aproximace na výslednou vzdálenost . . . . .	48
4.6	Druhá revize funkce Circular fit . . . . .	51
4.7	Načtení souborů a geometrické transformace . . . . .	53
4.8	Měření průměrné vzdálenosti mezi křivkami . . . . .	54
4.9	Hledání bodu s minimální vzdáleností od středu a měření délek . . . . .	56
4.10	Proložení kružnice a uložení výsledků . . . . .	58
4.11	Uživatelské rozhraní . . . . .	60

## Seznam zkratek

<b>ASCII</b>	American Standard Code for Information Interchange, 7 bitové kódování anglických znaků
<b>CD</b>	Compact disc, kompaktní disk, optické záznamové médium z osmdesátých let 20. století
<b>CP1250</b>	Kódovací sada češtiny používaná v Microsoft Windows
<b>\CR</b>	Carriage Return, znak návratu na začátek řádku
<b>CSV</b>	Comma separated value, textový formát dat, sloupce odděleny čárkou
<b>CT</b>	Computer-assisted tomography, počítačový tomograf
<b>DICOM</b>	Standard pro výměnu medicínských dat
<b>GMRES</b>	Generalized minimal residual method, metoda zobecněných reziduí, iterační metoda pro numerické řešení soustav lineárních rovnic
<b>JPEG</b>	Joint Photographic Experts Group, metoda ztrátové komprimace obrazových dat, rovněž i tvůrci této metody
<b>\LF</b>	Line Feed, znak nového řádku
<b>PDF</b>	Portable document format, přenositelný formát dokumentů, formát který by měl zajistit stejný vzhled dokumentů nezávisle na použitém softwaru
<b>Real-Time</b>	Označení časově kritických systémů vyžadujících odezvy v reálném čase
<b>RGB</b>	Red Green Blue, kódování barev pomocí červené, zelené a modré barevné složky
<b>ROI</b>	Region of Interest, oblast zájmu
<b>SI</b>	Sakroiliakální skloubení, spojení os sacrum a os ilium
<b>UTF-8</b>	Univerzální kódování všech používaných znaků, minimální délka znaku 8 bitů
<b>x86</b>	Zkratka pro architekturu procesorů kompatibilních s procesorem Intel 8086/80286/386/486

# 1. Úvod

Práce je součástí výzkumného projektu vedeného na spinální jednotce Krajské nemocnice v Liberci. Smyslem této práce tedy je vytvoření nástrojů k získávání dat o prostorových dispozicích dlah. Tyto data byla doposavad získávána obkreslováním na papír z monitoru. To mohlo vést k nepřesnostem a neumožňovalo to objektivní zhodnocení.

Při tvorbě těchto nástrojů tedy bylo zapotřebí se na tyto nedostatky zaměřit a zároveň příliš nezatížit zdravotnický personál složitostí obsluhy. Součástí práce rovněž musel být vhodný algoritmus pro vzájemné porovnání levé a pravé strany pánve, což byl jeden z požadavků od vedoucího práce, který toto potřeboval pro vyhodnocení hypotézy, že podle jedné ze stran lze zrekonstruovat druhou.

Vytvořené nástroje nejsou míněny jako zdravotnický prostředek<sup>1</sup> a proto nesmí být použity k léčení konkrétního pacienta<sup>2</sup>, ale pouze k ověření hypotéz ohledně vnitřních dlah pánve. Jako takové nepodléhají schvalujícímu procesu zdravotnických prostředků. Pokud by bylo rozhodnuto o jejich nasazení v procesu léčby pacientů, bylo by nutné je podstoupit patřičnému schvalovacímu procesu. Za tímto účelem byla tato práce doplněna o kompletní dokumentaci.

---

<sup>1</sup>Ve smyslu zákona 123/2000Sb. – O zdravotnických prostředcích (ve znění pozdějších předpisů)[22, zákon 123/2000Sb.]

<sup>2</sup>Dle definice v § 2, odstavcích 1 a 2, zákona 123/2000Sb.

## 2. K čemu to všechno je?

Jak již bylo zmíněno v úvodu, práce je řešením úkolu, jak vyhodnocovat různé parametry křivek z obrazových dat. Přičemž takto získaná data budou později použita jak pro výpočet tvaru průměrné pánevní dlahy, tak pro ověření předpokladu že pomocí křivky získané z jedné ze stran lze určit tvar druhé strany. Což by bylo možné uplatnit v případech zlomenin pánevního kruhu kdy je jedna ze stran neporušená.

Data jsou sbírána za účelem modelace nebo modifikace Omega dlahy, jak byla nazvána pánevní dlaha vyvinutá na pracovišti vedoucího práce ve spolupráci s firmou MEDIN, a. s. Dlaha dostala jméno podle svého tvaru kterým připomíná velké řecké písmeno  $\Omega$ . Na rozdíl od něj je ovšem většinou umístována obráceně (obrázek 2.1). Podrobnější informace o této dlaze lze získat ze zdroje [20].

Pokud se na základě takto získaných dat podaří nalézt dostatečně malý počet různých tvarů dlahy, tak by mohla být dlaha dodávána z výroby již před modelováním a tím by se snížila potřeba ručního ohýbání dlahy při zákroku a mohlo by dojít i ke zkrácení doby zákroku. Zároveň by však byly zachovány výhody sériové výroby jako jsou nižší výrobní náklady a náklady na logistiku.

V případě, že by se podařilo statisticky prokázat symetrii pánve, bylo by možné provádět u pacientů se zlomeninami pánve individuální modelaci dlahy na základě vyšetření počítačovým tomografem a následném vyhodnocení nepoškozené části. Což může vést k dalšímu urychlení výkonu a tím nižší zátěži pro pacienta, avšak se zde ztrácí výhody sériové výroby zmíněné v předchozím odstavci.

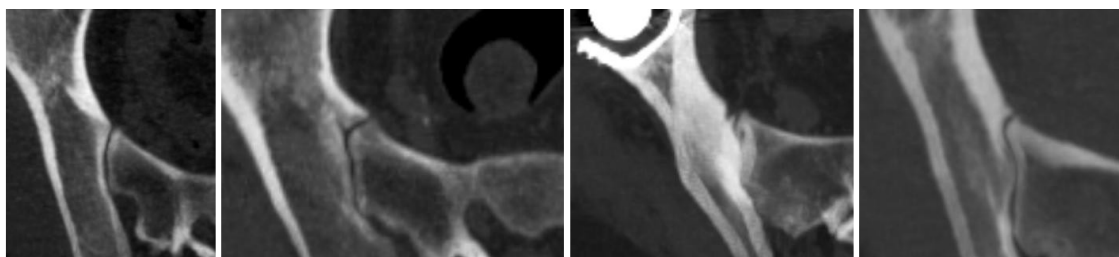


Obrázek 2.1: Omega dlaha (obrázek poskytnut vedoucím práce)



### 3. Vyhodnocování obrazových dat

V této kapitole bude podrobněji vysvětlena problematika získání křivky z řezu ve formátu JPEG případně z bitmapy. Je zde především využito základních metod počítačového vidění, avšak zde nejsou použité žádné optimalizace. Pro ně zde není uplatnění, protože se nejedná o takzvaný Real-Time systém, u kterého by bylo nutné docilovat velmi krátkých časů zpracování.



Obrázek 3.1: Rozdíly v SI skloubení mezi pacienty

Dalším rozdílem od technických aplikací, na který bylo nutné brát zřetel, je fakt, že lidské tělo se liší člověk od člověka. Tyto rozdíly spolu s proměnlivou kvalitou snímků (pohyb pacienta, odlišná nastavení CT), pak způsobují komplikace při automatické identifikaci objektů v řezu. Tyto rozdíly jsou patrné ve stydké sponě, v acetabulu i v SI skloubení. Vybraná SI skloubení jsou vyobrazena na obrázku 3.1, kde je vidět rozdíl jak ve tvaru, tak i v použitém měřítku.

Díky těmto komplikacím nebylo možné docílit zcela automatického režimu vyhodnocování, ale pouze režimu poloautomatického. Kde je po obsluze vyžadováno specifikování oblasti zájmu a výběr nejbližšího bodu ke středu acetabula.

Každý krok zpracování obrazu je vždy doplněn o výsledný obraz po aplikování daného filtru. Jako vzorový řez je použit jeden z iniciálních snímků (obrázek 3.2), které byly za účelem této práce (a na ní navazujícímu výzkumu) poskytnuty vedoucím práce. V rámci ochrany osobních údajů z nich byly vymazány údaje, které by umožňovaly identifikovat konkrétního pacienta.





Obrázek 3.2: Původní řez bez úprav

### 3.1 Prahování

Prvním krokem k získání křivky je prahování. Jedná o v zásadě jednoduchou operaci, při které je původně obraz v systému RGB nebo šedotónový převeden na binární. Postup probíhá tak, že jsou nejprve stanoveny takzvané prahovací hodnoty. Ty lze získat například z jasového histogramu, nebo empiricky metodou pokus-omyl. Z histogramu lze za pomoci některé z iteračních metod získat lokální minima výskytů jasových hodnot. Mé zkušenosti ze zpracovaných snímků ukazují, že ideální spodní hodnota prahového filtru se nachází v intervalu od prvního lokálního minima vpravo od globálního maxima histogramu. Rovněž se ukazuje, že maximální spodní hodnota prahového filtru nepřesahuje druhou pětinu jasového rozsahu. Za typickou hodnotu nastavení lze označit hladinu 75 při osmibitové barevné hloubce. Histogram s tímto nastavením filtru ukazuje obrázek 3.3 (zelená oblast je vyhodnocena jako kost, tedy s hodnotou jedna).



Obrázek 3.3: Histogram s nastavením prahového filtru

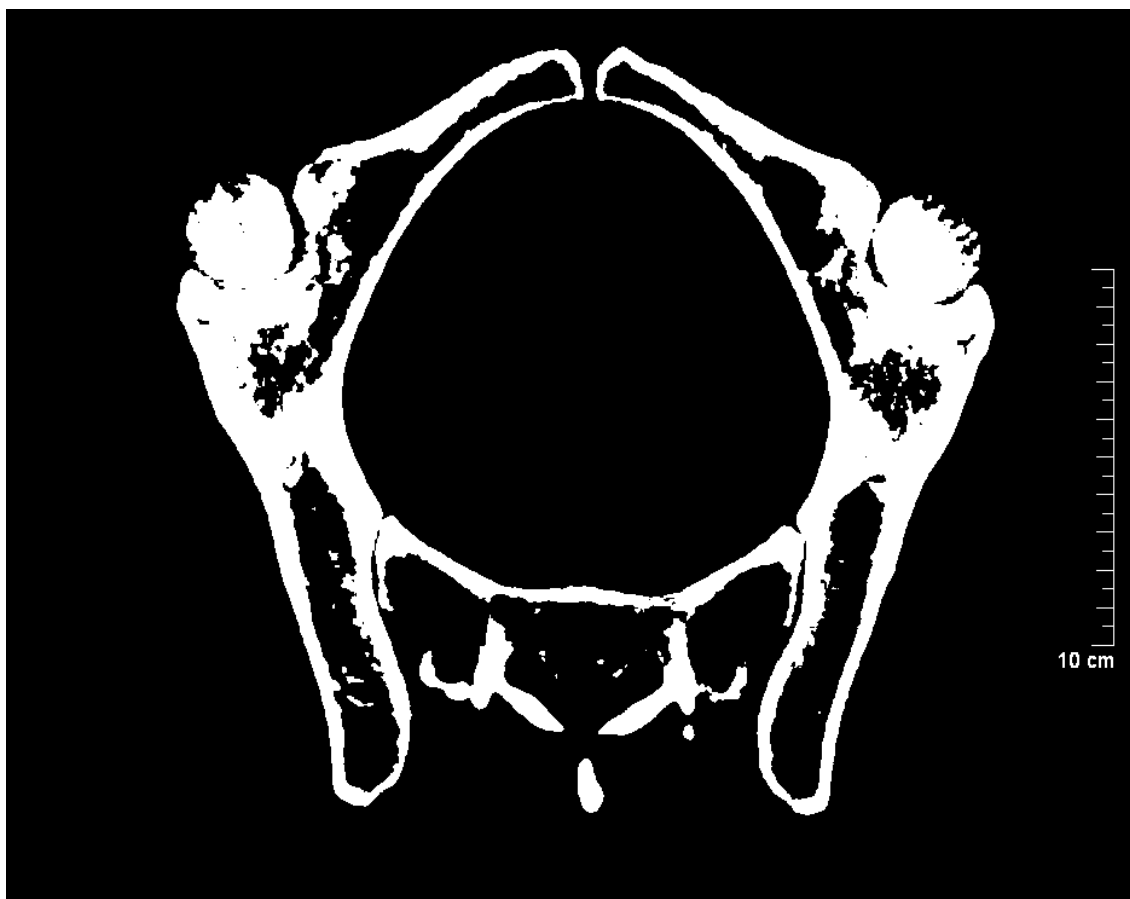
Další metodou, jak nastavit prahový filtr je za pomoci hypotézy, že relativní plocha kterou kost v řezu zabírá se více méně nemění. Tedy po prvotní identifikaci ideálního nastavení se sečte počet pixelů s hodnotou jedna po prahování a ten se vydělí celkovým počtem pixelů v obraze. Tím je získána relativní plocha kosti v řezu. Úlohou při zpracování dalších snímků je tento poměr udržet s co nejmenší odchylkou.

V programu je použit uživatelsky nastavitelný interval prahového filtru, ale nastavení se automaticky nemění. Tím, že rozdíly mezi snímky nemají na nastavení filtru takřka vliv, tak jsem se rozhodl automatický výpočet parametru neimplementovat. Toto opatření mělo za následek snížení jak výpočetního času, tak času potřebného na vývoj.

Dalším krokem je samotná aplikace nastaveného filtru. Tedy celý obraz je prohledán a pokud se hodnota jasu v pixelu nachází ve specifikovaném rozsahu, tak se tomuto pixelu přiřadí hodnota jedna, v opačném případě hodnota nula. Tímto je pak celý obraz převeden do binární podoby 3.4.

## 3.2 Hranová detekce

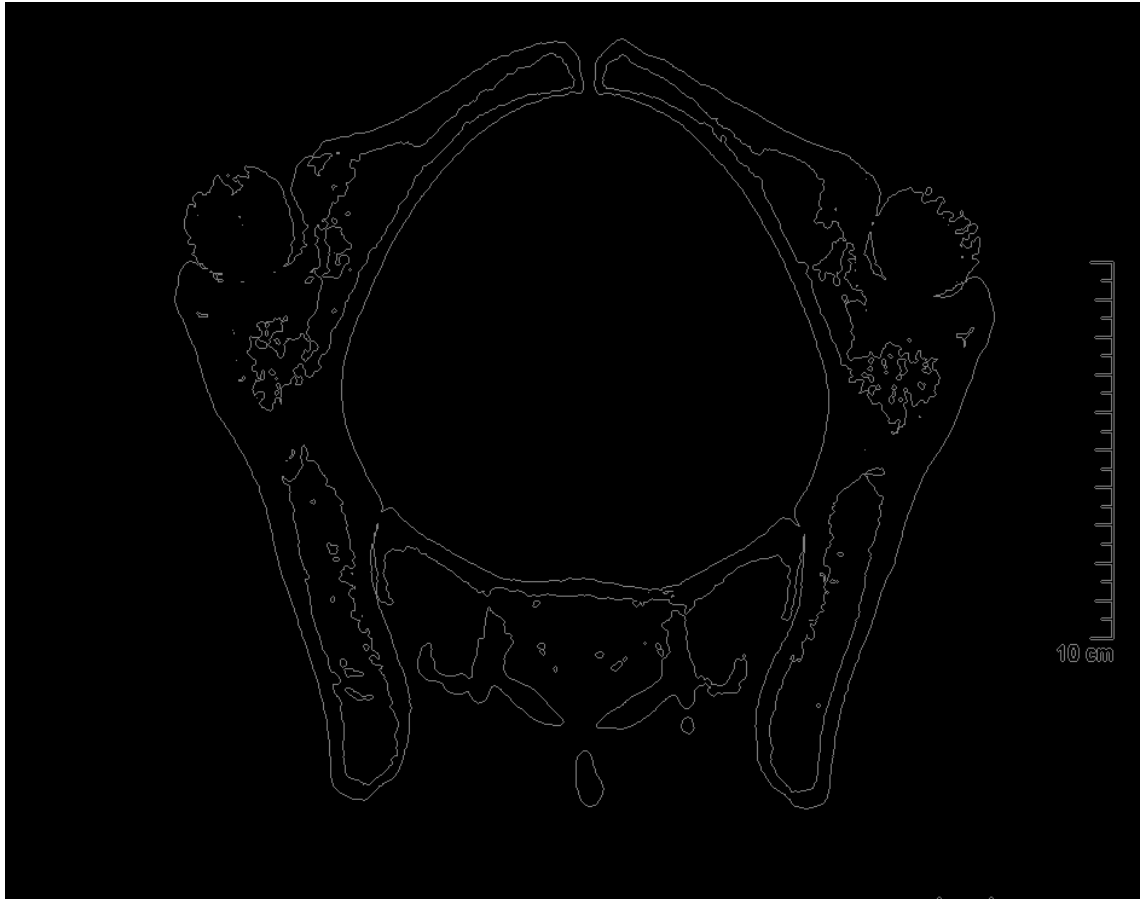
Hranová detekce je hlavní složkou tohoto programu. Už při iniciálním návrhu programu již bylo patrné, že rozhraní kost / ostatní tkáň bude možné nejlépe získat za pomoci derivace v obraze. Nebo přesněji diferenci jasových hodnot ve všech směrech, tedy pro osy  $x$  a  $y$  pro pixely (pro voxely i osa  $z$ ). Diference musí být použita místo derivace, jelikož obraz má konečný počet hodnot. To znamená, že



Obrázek 3.4: Řez po aplikování prahového filtru

obrazová funkce není diferencovatelná na celém intervalu obrazu (pokud není konstantní jasová hodnota konstantní). Dále bylo vzato do úvahy Marrova teorie [6], která říká, že druhá derivace (/difference) poskytuje lepší výsledky při detekci hran než derivace (/difference) prvního řádu.

Z konečného počtu bodů vychází i metoda za jejíchž pomoci je v obraze hrana detekována. Je zde totiž využíváno hlavně konvoluce. Funguje to tak, že je předem stanovena matice ve speciálním tvaru (tzv. konvoluční maska), taková pro kterou platí, že součet všech prvků matice je roven nule. To zajistí, že pokud je obrazová funkce konstantní, tak výsledný obraz po konvoluci má konstantní obrazovou funkci s hodnotou 0. Dalším krokem je posouvání této konvoluční masky po obraze a jejímu násobení s odpovídajícími body. Součet těchto součinů v každém bodě posunutí konvoluční masky je novou hodnotou tohoto bodu (rozhodující je typicky střed konvoluční masky). Obvykle jsou konvoluční masky voleny tak, že navíc v sobě obsahují i filtr pro potlačení šumu (dle [18], s. 83).



Obrázek 3.5: Řez po aplikování Laplaceova filtru

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (3.1)$$

$$\downarrow$$

$$grad(i)(x, y) = |i(x, y) - i(x + 1, y + 1)| + |i(x, y + 1) - i(x + 1, y)|$$

Jednou z nejjednodušších konvolučních masek je Robertsův operátor (3.1). Jeho jistou výhodou a zároveň i nevýhodou je malá velikost konvoluční masky. To znamená i malou výpočetní náročnost kvůli menšímu počtu matematických operací. Zároveň malá konvoluční maska přináší díky menšímu okolí i větší citlivost na šum, který zde není nikterak filtrován. To je hlavním důvodem proč není v praxi příliš nasazována, avšak může dobře posloužit pro vysvětlení principu použití konvolučních masek obecně. Pod konvolučními maskami 3.1 je rozepsán postup konvoluce pro získání obrazové funkce po hranové detekci  $grad(i)$  (dle [7], s. 2). Přičemž  $grad(i)(x, y)$  je vypočítaným elementem této funkce,  $i$  je původní obrazová funkce. Rozepsaný

vzorec lze získat například jako roznásobení konvoluční masky po prvcích s maticí obrazových bodů stejné velikosti (rovnice 3.2).

$$i_{2 \times 2} = \begin{bmatrix} i(x, y) & i(x, y+1) \\ i(x+1, y) & i(x+1, y+1) \end{bmatrix}$$

$$i_{3 \times 3} = \begin{bmatrix} i(x-1, y-1) & i(x-1, y) & i(x-1, y+1) \\ i(x, y-1) & i(x, y) & i(x, y+1) \\ i(x+1, y-1) & i(x+1, y) & i(x+1, y+1) \end{bmatrix} \quad (3.2)$$

Složitější, ale zato používanější maskou je Lapaceův operátor. Ten aproximuje druhou derivaci, je invariantní vůči otočení, vytváří dvojistou odezvu na tenké linie a je lineární (dle [7] a [12], s. 87). Příklady možných konvolučních masek jsou uvedeny v rovnici 3.3 a obrázek po aplikaci konvoluce má číslo 3.5. Jak je na tomto obrázku patrné, tak jediné co zbývá pro získání finální křivky je specifikování oblasti zájmu a výběr středu soustavy.

$$h_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, h_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.3)$$

Mimo těchto konvolučních masek existují pochopitelně i další. Například existují i varianty Lapaceova operátoru s vyšší váhou směrem k rozhodnému bodu masky, které již nejsou invariantní k otočení. Dále bych mohl jmenovat operátor podle Prewittové, Sobelův operátor, Robinsonův operátor a Kirschův operátor. Všechny tyto operátory aproximují první derivaci, nejsou invariantní na otočení, jsou nelineární a konvoluce se u nich provádí pro osm směrů přičemž výsledkem je největší absolutní hodnota.

Krom konvoluce existuje i poněkud jiný přístup, který bych zde měl zmínit. U něho je použito binárních morfologických operací, konkrétně eroze a následné odečtení od původního obrazu. Eroze se provádí tak, že se nejdříve určí tvar okolí, které pokud bod má, tak se nahradí jedničkou, pokud takové okolí nemá, tak se nahradí v novém obrazu nulou. Po odečtení od originálu jsou získány hrany obrazu s velikostí a orientací danou zvoleným okolím.

Všechny operace zmíněné v této kapitole jsou běžně používané v průmyslu v oblasti počítačového vidění. Jsou výpočetně nenáročné a snadno paralelizovatelné. To

umožňuje jejich používání i v Real-time systémech, kde je výpočetní čas kritický. Zde však nebylo nutné porovnávat, či dokonce pozměňovat výpočetní algoritmy, jelikož zde čas natolik kritický není. V programu byl proto ponechán výchozí algoritmus hranové detekce, který jakožto součást firemního tajemství není přesně znám. Avšak podle [12] se lze domnívat, že se jedná o Laplacián  $h_8$  z rovnice 3.3.

### 3.3 Výběr křivky

Jak již bylo zmíněno na začátku kapitoly, kvůli přílišným rozdílům mezi pacienty nebylo možné dosáhnout automatického vyhodnocení. Hlavní kámen úrazu nastal právě u výběru oblasti zájmu, potažmo křivky. Standardní hledání tvarů, jako se používá v průmyslových aplikacích by zde nejspíše selhalo, nebo by při nejmenším mělo velmi nízkou úspěšnost. Proto byla zvolena varianta, kdy oblast vybírá uživatel a má tak zároveň i možnost kontroly správnosti výběru. Uvažovaná křivka se mu totiž označí v obraze modrou barvou (podrobněji se tím budu zabývat v kapitole Uživatelské rozhraní).

Z programového hlediska je pro získání křivky potřeba specifikovat oblast zájmu (v kódu označeno jako ROI), prahovou hodnotu hrany, minimální délku křivky, směr vyhledávání, typ výběru a v neposlední řadě samotný předzpracovaný obraz. Oblast zájmu musí specifikovat uživatel tak, aby obdélník výběru obsáhl celou uvažovanou křivku, aby se nedotýkala okraje výběru jinde než na koncích a tak aby konce křivky byly jeden před SI skloubením a jeden před stydkou sponou. Prahová hodnota hrany reprezentuje minimální hodnotu, kdy program vyhodnotí změnu jasu jako hranu. V našem případě, kdy již používáme předzpracovaný binární obraz je to hodnota jedna.

Minimální délka křivky se ukázala jako dosti důležitý parametr. Slouží totiž k odfiltrování menších oblastí, které se občas v řezu objeví a to pak vede k vyhodnocení úplně jiné křivky. Při ladění bylo zjištěno, že spolehlivě funguje parametr s hodnotou sto. V případě pokud by byla vyhodnocována špatná křivka, je v programu ponechána možnost uživatelské změny za běhu.

Směr vyhledávání se naopak ukázal jako méně významný parametr. Ve zkratce jde o to jakým směrem bude v ROI křivka vyhledávána. Správné je jakékoliv nastavení, při kterém vyhledávání postupuje směrem od středu obrazu k okraji. Nevhodná volba způsobí pouze vyšší náročnost na přesnost specifikování ROI. Typ výběru

umožňuje zvolit jestli bude vybrána nejbližší křivka od startu hledání, nejdelší křivka nebo křivka s největším jasovým gradientem. Přičemž poslední jmenovaná nepřípadá v úvahu kvůli binárnímu obrazu (tedy gradient je všude stejný). Mezi prvními dvěma lze libovolně volit a záleží na preferencích každého uživatele.

Posledním údajem který je v tomto kroku nutné dodat je střed acetabula. Tedy přesněji bod na křivce s minimální vzdáleností od středu acetabula. Na základě konzultací s vedoucím práce bylo rozhodnuto, že vhodnou aproximací tohoto bodu bude průsečík spojnice přilehlého středu hlavice femuru a protějšího SI skloubení s křivkou. Tento úzus umožňuje minimalizovat chyby způsobené subjektivním hodnocením obsluhy. Osobní zkušenost mi ukázala, že přesná identifikace tohoto bodu může být pro méně trénované oko přinejmenším nejednoznačná.

### 3.4 Seznámení s prostředím LabView

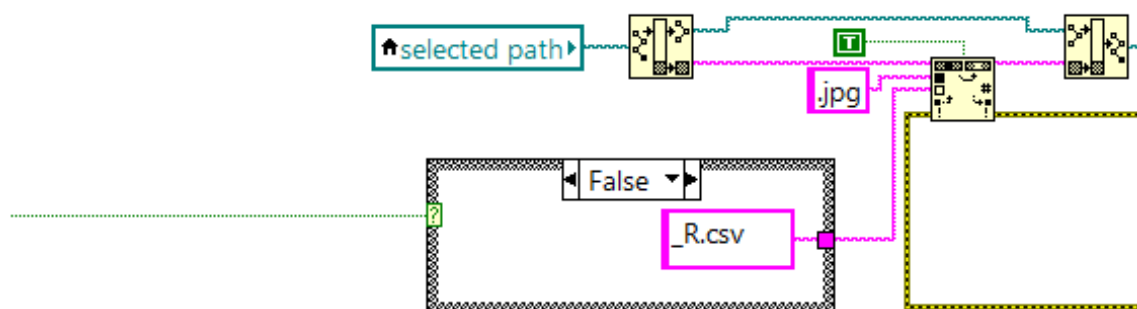
V předchozích kapitolách jsem stručně probral kroky spojené se získáním křivky rozhraní kost/ostatní tkán. Teď je na čase popsat vývojové prostředky použité k tvorbě samotného programu. Tedy při tvorbě programů jsem používal hlavně prostředí LabView s modulem Vision Assistant od firmy National Instruments ve verzi 2012. Důvodu k použití tohoto vývojového prostředí bylo hned několik. V první řadě v něm pracuje konzultant práce Ing. Hančil. Což se ukázalo jako nesporná výhoda, protože to umožňovalo mnohem efektivnější a rychlejší spolupráci při řešení problémů s implementací.

Druhou výhodou byla určitá abstrakce od běžných prvků objektově orientovaného programování, jako tvorba vlastních tříd, dědičnosti, distribuce událostí nebo vytváření datových struktur. Jelikož se jedná o grafický programovací jazyk, tak je zde možné soustředit se více na řešení úkol než na jeho formální zápis. Tedy programování v tomto jazyce spočívá v propojování již naprogramovaných funkčních bloků, ovládacích prvků, konstant a tak podobně.

Prostředí LabView má pochopitelně i nějaké nevýhody. Osobně za největší z nich považuji svázanost s platformou Microsoft Windows a architekturou x86. Prostředí existuje i ve variantě pro MacOS X a Linux, ale na těchto platformách nedosahuje ani zdaleka plné funkcionality. Díky tomu jsou pak i vytvořené programy svázané s Windows. Na druhou stranu vytvářet kvůli multiplatformitě celý program například v C++ by časovou náročností neodpovídalo výsledkům. Zvláště pokud použití na

alternativních platformách nebylo součástí zadání.

Teď k samotnému prostředí. Při otevření projektu se objeví dvě okna. Jedno pro tvorbu vnitřní struktury programu a jedno pro návrh uživatelského rozhraní. Při přidání prvku do jednoho z oken se vytvoří i korespondující prvek ve druhém z nich (pokud existuje, například matematické operace nemají uživatelské rozhraní). Prvky se přidávají pomocí pravého tlačítka myši a jejich nabídka záleží na zakoupených rozšiřujících modulech. Typická podoba kódu je vidět například na obrázku 3.6.



Obrázek 3.6: Ukázka kódu v LabView (změna cesty k souboru)

Různé barvy čar na obrázku reprezentují různé typy proměnných. Kupříkladu zelená tečkovaná čára reprezentuje proměnnou typu binárního čísla. Tedy může nabývat pouze hodnot jedna (pravda, TRUE, T) nebo nula (nepravda, FALSE, F). Konstanta logické jedničky je konec konců na obrázku také vidět (bílé zarámované T v zeleném poli). Modrozelená odpovídá cestě k souboru, růžová řetězci a žlutočerná značí chybové zprávy. Zde zobrazené zarámování odpovídá konstrukci switch/case z konvenčních jazyků. LabView neodlišuje mezi konstrukcí obyčejné podmínky (if v konvenčních jazycích) a switch/case. Jediný rozdíl je větší počet stavů a jiný typ vstupní proměnné.

Zobrazená část programu mění část cesty k obrazu řezu a vytváří z ní dvě nové pro soubor s daty křivky. Podmínka pro výběr se realizuje mimo zobrazený výřez 3.6, jedná se však pouze o porovnání souřadnice  $X$  středu acetabula s polovinou šíře obrazu. Tedy určení jaká strana se právě vyhodnocuje. Tímto opatřením je zajištěno, že pokud jsou vyhodnocovány obě strany, tak druhá z vyhodnocovaných stran nepřepíše data první. Zároveň je formou názvu souboru řezu zajištěn unikátní identifikátor aniž by bylo nutné vyhodnocovat text na snímku a ten pak patřičně anonymizovat aniž by se ztratila jednoznačnost. Nehledě na možnost zpětně identifikovat eventuální špatně vyhodnocený snímek.



## 3.5 Implementace programu

Program je koncipován sekvenčně do šesti kroků. V této části jsou všechny kroky podrobněji popsány, ty komplikovanější jsou navíc doplněny o obrázek zdrojového kódu. Nejedná se však o kompletní program, ten je k dispozici v příloze a v dokumentaci k programům.

### 3.5.1 První krok – načtení obrazu

První část programu zajišťuje výběr, načtení obrazu z formátu JPEG a zobrazení náhledu. Krom toho je na obrázku 3.7 vidět zablokování ovládacích prvků, vypnutí extrakce křivky a snímání souřadnice středu acetabula, zaznamenání počtu pixelů v ose  $X$ , smazání výběru oblasti zájmu a zastavení programu pokud uživatel nevybere odpovídající soubor.

K zablokování tlačítka „OK“ je přikročeno, aby nedošlo k exportování křivky, aniž by byly nastaveny souřadnice středu acetabula. To by pak způsobilo problémy při pozdějším zpracování dat, protože se dále předpokládá, že tento bod má souřadnice  $[0; 0]$ . Tudíž by nebylo možné křivku rozdělit na frontální a dorzální část a ty zpracovávat samostatně. Jedná se tak zároveň i o prvek kontroly uživatele.

Vypnutím extrakce křivky a snímání středu acetabula je zabráněno uživateli aby mohl vyexportovat křivku aniž by zkontroloval nastavení prahového filtru a filtru pro potlačení malých objektů. Opomenutí této kontroly by mohlo vést například k vyhodnocování vnitřku kosti místo vnějšího obrysu, pokud by byly okraje méně kontrastní. Ačkoliv se nejedná o nikterak pravděpodobnou chybu při standardním nastavení, tak k ní teoreticky může dojít pokud by bylo zpracováváno více snímků s velmi odlišnou kvalitou. Jedná se tak více méně o preventivní opatření pro eliminaci opomenutí uživatele.

Zaznamenání počtu pixelů v ose  $X$  je důležité pro pozdější vyhodnocení, která strana pánve je vyhodnocována. To je v pozdějších krocích programu provedeno porovnáním  $X$ ové souřadnice středu acetabula s polovinou rozměru osy  $X$ . Pokud je souřadnice středu menší než polovina snímku, jedná se o pravou stranu pánve (na snímku vlevo), pokud je naopak větší, jedná se o levou stranu pánve (na snímku vpravo).

Vymazání oblasti zájmu (ROI) taktéž důležité pro eliminaci chyb obsluhy. Pokud by



k vymazání nedošlo, tak by se mohlo stát, že by uživatel neurčoval ROI pro každý snímek, ale namísto toho ponechával nastavení ze snímku minulého. To by vzhledem ke značné rozdílnosti jednotlivých snímků mohlo vést minimálně zkresleným údajům o délkách jednotlivých úseků. Na vině by v takovém případě bylo nepřesné zaměření konců křivek.

### 3.5.2 Druhý krok – filtrování obrazu

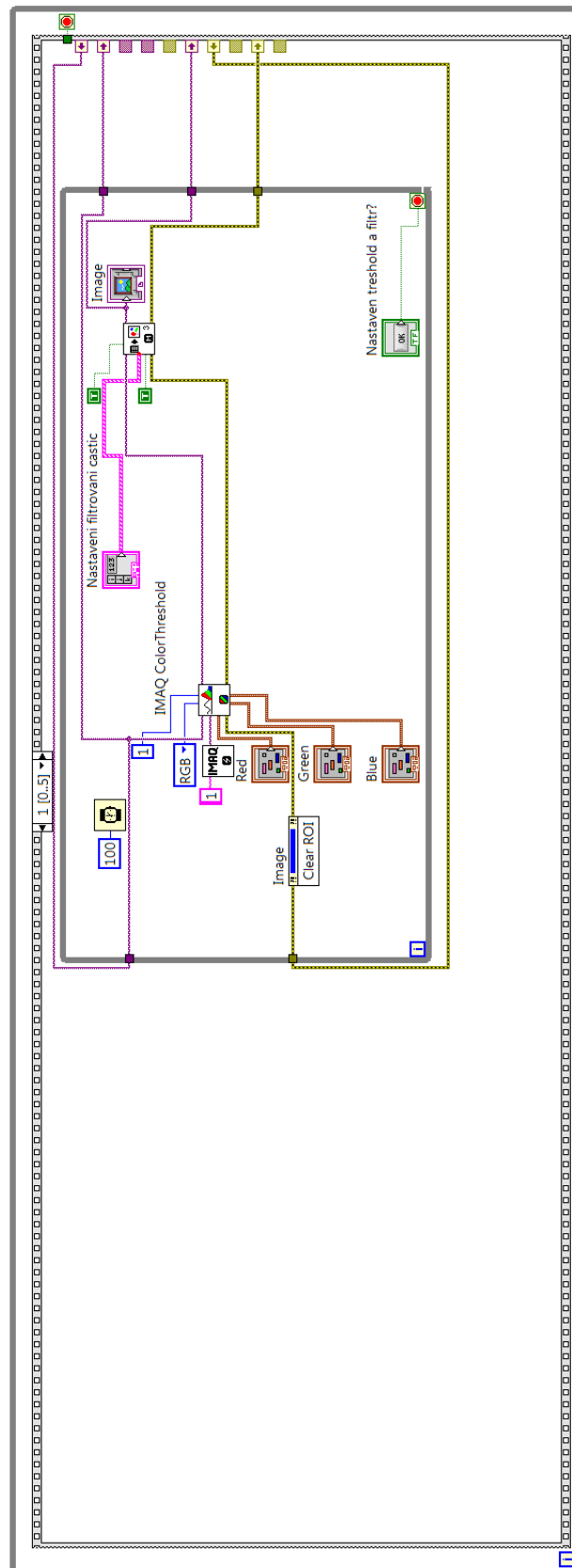
V druhém kroku je realizováno filtrování částic a prahový filtr (obrázek 3.8). Kvůli možnosti manuálně měnit nastavení je celý tento cyklus umístěn do tzv. „while smyčky“. Přičemž její ukončení je vázáno na stisknutí tlačítka „Next“. Provádění smyčky je kvůli úspoře systémových prostředků omezeno na 100 ms. Avšak toto omezení nemá téměř žádný vliv na plynulost uživatelského rozhraní.

Uvnitř smyčky je vždy nejdříve vynulována oblast zájmu. Toto opatření zabraňuje uživateli vybrat oblast aniž byly nastaveny filtry. Přesněji řečeno, pokud se mu podaří vybrat oblast zájmu, tak je mu nejpozději do 100 ms smazána. Tedy uživatel nemůže zapomenout minimálně potvrdit nastavení filtrů.

Dále je ve smyčce vykonáno prahování obrazu podle uživatelského nastavení. To je tvořeno samostatnými spodními a horními hranicemi pro všechny tři barvy. Jedná se o obdobu postupu uvedeného v kapitole Prahování, avšak zde rozdělenou do tří složek (RGB). Rozdělení je výsledkem ladění v raných fázích vývoje programu, kdy jsem předpokládal lepší dosažené výsledky pokud bude možné nastavit složky samostatně. Tento předpoklad se později ukázal jako nepodložený. Je tedy možné nahradit použitý prahový filtr převodem obrazu do šedotónového formátu a ten poté prahovat podle hodnoty jasu (jak bylo uvedeno ve zmíněné kapitole).

Poté je aplikován filtr částic. Ten má za úkol odstranit částice menší než je určitý, uživatelsky definovatelný počet pixelů. Tento filtr se ukázal jako velmi užitečný v případech, kdy se v obraze vyskytuje šum, artefakty menšího rozsahu a kdy pacientům byly podány kontrastní látky. Hlavně poslední zmíněný případ by jinak mohl způsobit pozdější chyby v detekci křivky, nebo jí minimálně ztížit formou obtížnějšího určování oblasti zájmu.

Jakmile je provedena filtrace částic je výsledný obraz překreslen v hlavním pracovním okně. To umožňuje průběžnou kontrolu nastavení ještě před jejich konečným potvrzením. Po potvrzení totiž program přejde do dalšího kroku a parametry filtrů



Obrázek 3.8: Prahování obrazu a filtrování částic

tak již není možné měnit.

### 3.5.3 Třetí krok – odblokování extrakce

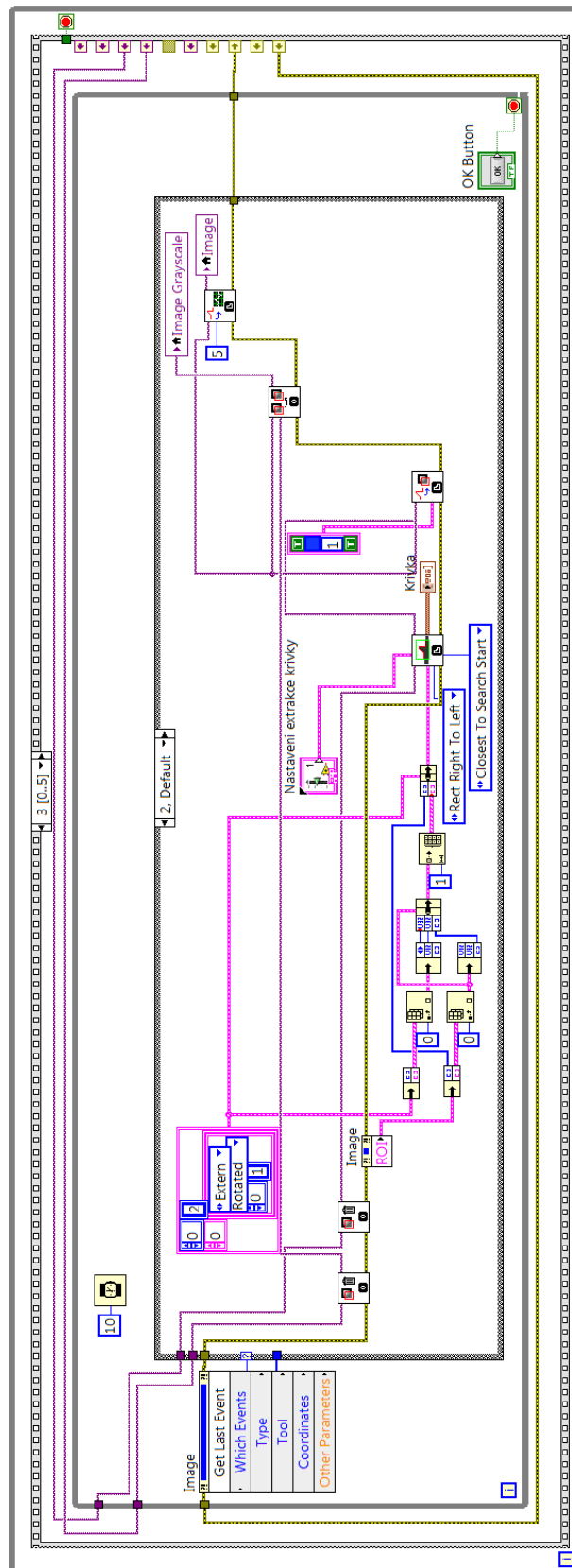
Ve třetím kroku je povolen výběr souřadnic středu acetabula a extrakce křivky. Tento krok zde není vyobrazen, jedná se však pouze o propojení dvou funkčních bloků chybovým vodičem a přiřazení konstanty „Enabled“ do obou bloků. Je vhodné zde zmínit, že i když by zde nebyl chybový vodič vlastně potřeba, tak zajišťuje i jednu zajímavou funkci. V LabView se totiž vykoná funkční blok až když jsou k dispozici signály ze všech vstupů. Tyto signály mohou mít i nulovou hodnotu, ale musí k bloku dorazit. Příkladem může být chybový vodič, tím v ideálním případě proudí prázdná data, ale ty jsou následujícímu bloku předána až po vykonání předchozího bloku. Toho je v programu několikrát využito k určení pořadí vykonávání funkčních bloků.

### 3.5.4 Čtvrtý krok – extrakce křivky

Čtvrtá část zajišťuje zachytávání událostí z hlavního pracovního okna. Těmito událostmi se rozumí hlavně událost s číslem dva (kreslení) a událost osm (dvojitě kliknutí). Podle čísla události je provedená odpovídající akce ve struktuře „case“. Na obrázku 3.9 je zobrazeno zpracování kreslení, dvojitě kliknutí zde zobrazeno není.

Každých 10 ms je snímána poslední událost v okně. Pokud není zaznamenána událost dvojitě kliknutí, je provedena vyobrazená část kódu. Tedy nejdříve se vymažou překrývající vrstvy z náhledu a pracovního okna. Pak se sestaví popis oblasti zájmu, který je následně předán k separaci křivky z filtrovaného obrazu. Odtud jsou získány souřadnice bodů křivky. Následující funkční bloky vyznačují vyhodnocovanou křivku do obou oken.

Pokud je naopak zachycena událost dvojkliku, tak dojde k zaznamenání souřadnic a odblokování tlačítka „OK“. Tím je uživateli umožněn export dat křivky a zároveň jsou souřadnice středu acetabula nad tímto tlačítkem zobrazeny. To umožňuje zároveň kontrolu souřadnic vůči aktuální poloze kurzoru v pracovním okně. Ta je spolu s ostatními údaji na spodní hraně okna. Po stisknutí tlačítka „OK“ program opustí „while smyčku“ a dále již pokračuje bez interakce s obsluhou.



Obrázek 3.9: Extrakce křivky

### 3.5.5 Pátý a šestý krok – předzpracování a uložení

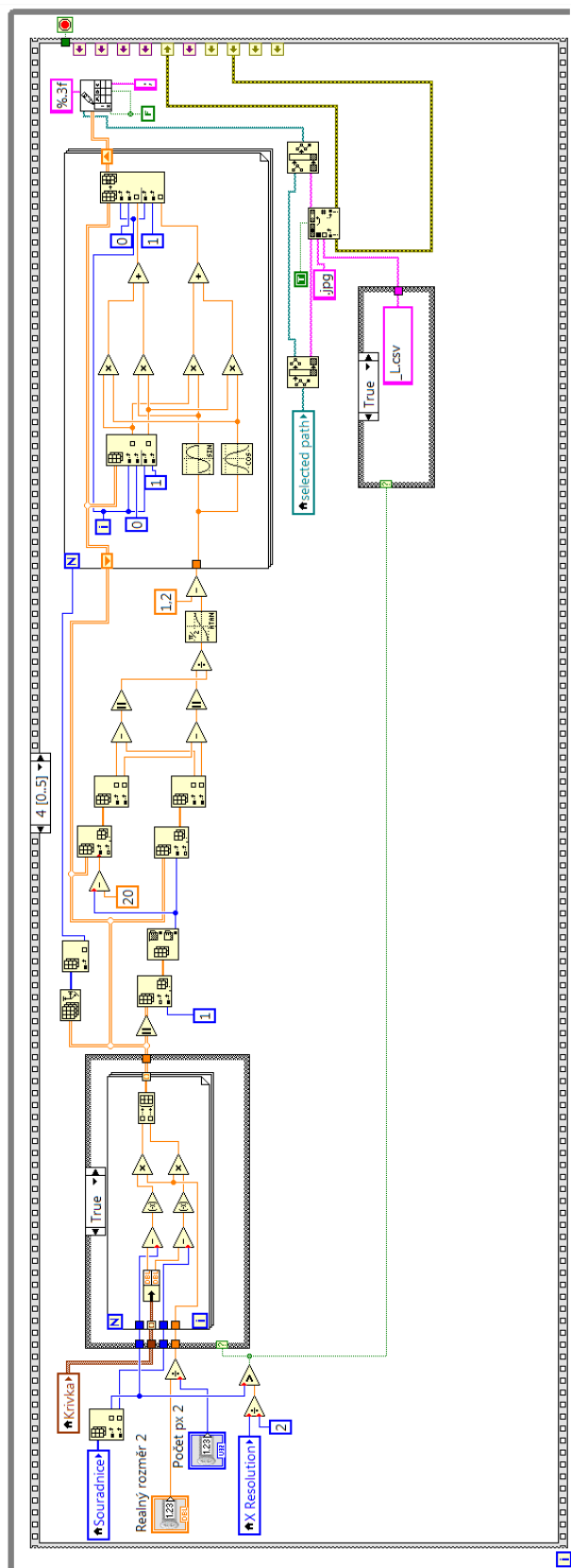
V předposledním kroku je zjištěna strana která byla zaznamenávána, křivka je zde převáděna na reálné rozměry, otáčena a posunuta středem acetabula do středu souřadného systému. Dále je vytvořena cesta k souboru do kterého se budou data zapisovat a následně se provede samotný zápis.

Jak je vidět na obrázku 3.10, tak nejprve dojde k rozkladu souřadnic středu acetabula na dvě samostatná čísla (souřadnice  $X$  a  $Y$  v pixelech). Současně je porovnáním souřadnice středu acetabula a poloviny rozměru obrazu na ose  $X$  zjištěno, která strana pánve se právě zpracovává. To ovlivňuje zda bude výsledná křivka otáčena podle osy  $Y$ , aby jí bylo možné později porovnat s křivkou z druhé strany.

Pokud je zpracovávána pravá strana pánve (kód na obrázku 3.10, tak dochází pouze k otáčení podle osy  $X$ . K tomuto je přikročeno kvůli rozdílům v chápání souřadného systému v obraze jak je uložen v počítači a mezi souřadným systémem jak jej vnímá člověk. Zatímco v počítači je bod s nulovými souřadnicemi vlevo nahoře, tak člověk předpokládá nulu souřadného systému vlevo dole. Zároveň v počítači je kladný směr na ose  $Y$  směrem dolů, tak člověk předpokládá kladnou změnu souřadnic směrem k hornímu okraji obrazu.

$$p = \frac{[realny\ rozměr]}{[rozměr\ v\ pixelech]} \quad (3.4)$$

Přepočet na reálné rozměry je prováděn pomocí přepočetniho koeficientu. Jeho výpočet je vidět v rovnici 3.4, kde oba tyto parametry zadává uživatel v předchozí části programu podle měřítko ve snímku. Reálný rozměr tedy bývá 10 cm, ale rozměr v pixelech je proměnlivý v závislosti na rozlišení snímků.



Obrázek 3.10: Otočení a uložení křivky



$$\begin{aligned}
\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -s_x \\ 0 & 1 & -s_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&\downarrow \\
\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} p & 0 & -ps_x \\ 0 & -p & ps_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&\downarrow \\
x' &= p \cdot x + 0 \cdot y - p \cdot s_x \cdot 1 \\
y' &= 0 \cdot x - p \cdot y + p \cdot s_y \cdot 1 \\
1 &= 0 \cdot x + 0 \cdot y - 1 \cdot 1 \\
&\downarrow \\
x' &= px - ps_x \\
y' &= -py + ps_y \\
&\downarrow \\
x' &= p(x - s_x) \\
y' &= -p(y - s_y)
\end{aligned} \tag{3.5}$$

Celkový přepočet souřadnic včetně posunutí a posunutí do nuly lze nalézt v rovnicích 3.5. Kde  $x'$  a  $y'$  jsou nové souřadnice bodu v reálném rozměru,  $x$  a  $y$  jsou staré souřadnice v pixelech,  $s_x$  a  $s_y$  jsou souřadnice středu acetabula v pixelech a  $p$  je přepočtení koeficient měřítka. Jedná se o kombinaci změny měřítka a translace (více se touto problematikou zabývám v kapitole Translace) pro pravou část pánve. Otočení pro levou část je možné docílit přidáním znaménka mínus k přepočtenímu koeficientu  $p$  v prvním řádku první matice.

Sestavení cesty k souboru pro export dat jsem již zmiňoval v kapitole Seznámení s prostředím LabView. Ve zkratce jde o to, aby bylo možné jednoznačně identifikovat pacienty a zároveň, aby data jedné strany pánve nepřepisovala data druhé strany. Pro splnění první podmínky jsem zvolil označení dat názvem souboru řezu. Tím pádem není nutné číst data ze snímku a předchází se tak možnému zdroji problémů. Splnění druhé podmínky lze řešit dvěma způsoby. Prvním by bylo oddělit strany ve vnitřní struktuře souboru. To by však znamenalo komplikace v případech, kdy by se vyhodnocovala pouze jedna strana, nebo v případech opakovaného měření. Daleko jednodušší a bezpečnější způsob spočívá v použití dvou souborů, každý pro jednu stranu. Vyšší bezpečnost je způsobena kontrolou, že byly vyhodnoceny obě

strany a zároveň zde zůstává možnost opakování chybného měření pouze na jedné straně.

K odlišení stran je použito řetězců „\_R.csv“ pro pravou stranu a „\_L.csv“ pro levou stranu. Přičemž jimi je nahrazena původní přípona řezu „.jpg“. Vyhodnocování kritérium je použito stejné jako u přepočtu souřadnic. Po vytvoření cesty je provedeno uložení do textového souboru. Jako oddělovač souřadnic je použit středník, jako desetinný oddělovač čárka, oddělení bodů zajišťuje nový řádek pomocí znaků `\CR` a `\LF` a čísla jsou zapsána jako text se třemi desetinnými místy. Tím je dosažen formát standardního souboru typu CSV v adaptaci pro české Microsoft Windows a lze tedy pro čtení dat použít běžné tabulkové procesory (například LibreOffice Calc nebo Microsoft Excel).

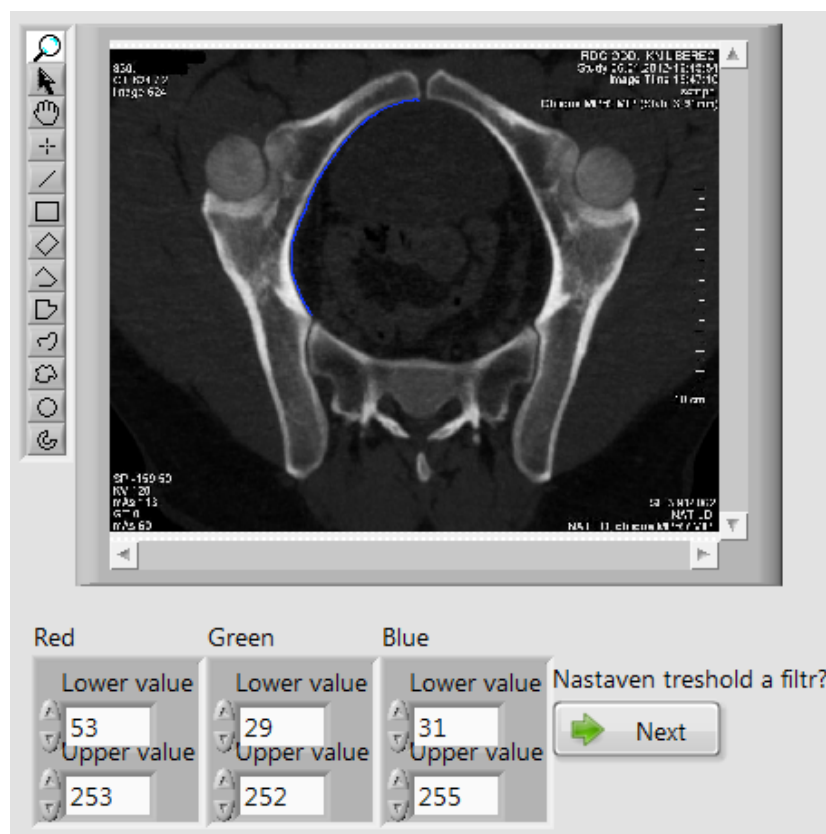
V posledním kroku jsou pak pouze vyčištěny chybové zprávy a vyhodnoceno tlačítko posledního snímku. Pokud bylo stisknuto, tak po uložení dat dojde k vypnutí programu. Pokud ne, dojde k otevření souborového dialogu pro výběr dalšího snímku a program běží od začátku.

## 3.6 Uživatelské rozhraní

Uživatelské rozhraní odpovídá sekvenčnímu běhu programu, přičemž stačí aby uživatel postupoval při ovládání zleva doprava a neměl by narazit na větší problémy. Po startu programu se nejdříve objeví klasický souborový dialog. V něm uživatel musí vybrat vyhodnocovaný snímek ve formátu JPEG a příponou „.jpg“. Stojí za zmínku, že výstupní data budou uložena do stejného adresáře odkud byla získána data vstupní a budou mít první část názvu společnou (rozšířenou o „\_R.csv“ nebo „\_L.csv“ viz výše).

Po úspěšném načtení souboru se vlevo nahoře objeví náhled původního snímku. V tuto chvíli je od uživatele požadována kontrola nastavení prahového filtru (obrázek 3.11) a filtru pro odstranění drobných částic. První z jmenovaných je aktivní vždy a je k nastavení poměrně benevolentní. K problémům s implicitním nastavením však může docházet u málo kontrastních snímků. Uživatel by si měl dát hlavně pozor, aby nedocházelo k rozpoznání okolních struktur pánve jako kosti a zároveň aby již nedocházelo ke změnám na rozhraní pánevního kruhu.

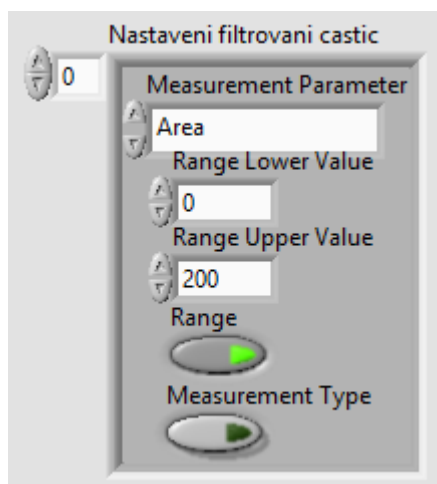
Filtr pro odstranění drobných částic není vždy potřeba, takže se může stát, že nebude ve výchozím stavu zapnutý. Pokud však bude uživatel zpracovávat velmi zašuměný



Obrázek 3.11: Náhled a nastavení prahování

obraz má možnost filtr aktivovat. Pokud je nastavení filtru neaktivní lze ho přidat pomocí položky „Insert Element Before“ z kontextové nabídky nastavení filtru. Dále doporučuji vybrat u „Measurement Parameter“ možnost „Area“, dolní hranici nechat na hodnotě nula, horní hranici nastavit minimálně na 5 a zaškrtnout volbu „Range“ (příklad nastavení je na obrázku 3.12). Během nastavování filtru se může stát, že z pracovního okna zmizí obraz. To je způsobené nevhodným nastavením filtru nebo nezaškrtnutou volbou „Range“, kdy filtr zobrazuje části, které by se měly z obrazu odstranit. Uživatel si musí dát pozor, aby před potvrzením nastavení viděl v pracovním okně opravdu ten obraz, který chce vyhodnocovat. K tomuto nastavení se lze vrátit pouze po restartu programu nebo načtení nového snímku.

Po potvrzení nastavení filtrů tlačítkem „Next“, přichází na řadu práce v hlavním okně. Konkrétně je nutné identifikovat střed acetabula a vymežit oblast zájmu. Osobně doporučuji nejdříve přepnout v panelu nástrojů na úsečku (na obrázku 3.13 vlevo nahoře, pátá položka shora), tou si vyznačit spojnici středu hlavičky femuru a protějšího SI skloubení. Poté přepnout zpět na kurzor a dvakrát kliknout na průsečík této úsečky a uvažované křivky. Tím dojde k označení středu acetabula, což

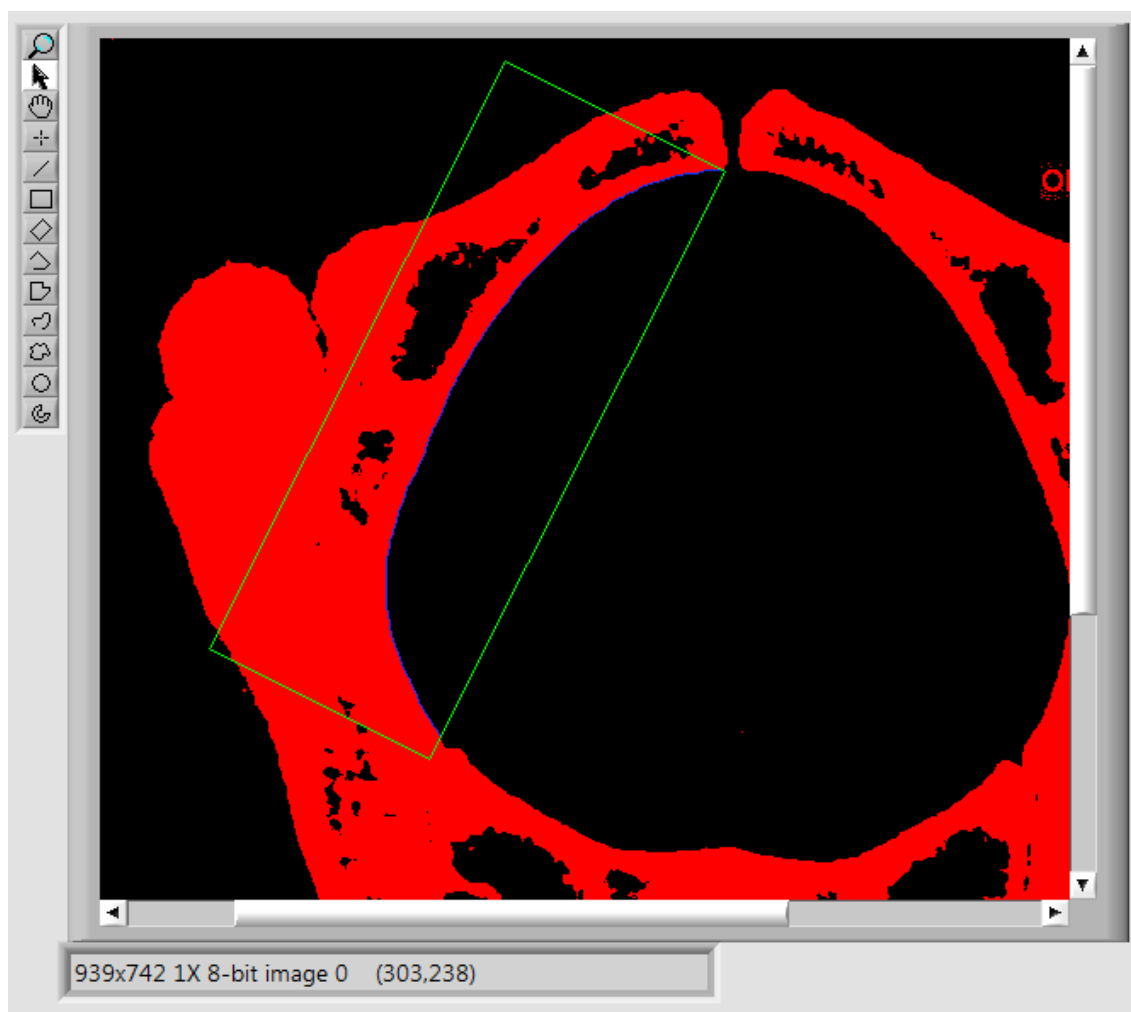


Obrázek 3.12: Příklad nastavení filtru částic

je indikováno nenulovými souřadnicemi nad tlačítkem „OK“ a jeho odblokováním. Následně zvolit ikonu otočeného obdélníku (sedmá shora, z ostatních tvarů program křivku nevyhodnocuje). Táhnutím v odraze je pak vytvořena oblast zájmu, kterou je možno pomocí kurzoru dále upravovat. Křivka je v obraze zvýrazněna modrou barvou stejně jako v náhledu.

Pokud program vybral správnou křivku, tak máte vyhráno a stiskem tlačítka „OK“ se křivka uloží a pokud nebylo předtím stisknuto tlačítko „Toto je poslední snímek“, tak program začne zase od začátku. Pokud však takové štěstí nemáte, zbývá několik možností jak situaci řešit. První možností je upravit oblast zájmu tak, aby v ní byla ideálně pouze křivka kterou chcete. Když to nezabere nebo to z nějakého důvodu nejde, je zde možnost změnit minimální délku křivky. To zajistí, že křivky nesplňující toto kritérium budou ignorovány, pozor však aby nebyla minimální délka nastavená na příliš vysoké číslo, pak by mohly být ignorovány všechny křivky.

Poslední možností je změna režimu vyhledávání křivky z „Closest To Search Start“ na „Maximal Length“ či opačně. Obvykle má hledání křivky s maximální délkou větší úspěšnost, zvláště pokud není v oblasti zahrnuta obvodová křivka.



Obrázek 3.13: Výběr oblasti zájmu

## 4. Analýza získaných křivek

V této části se práce zabývá zpracováním již dříve získaných křivek. K tomuto účelu jsem naprogramoval samostatný program. K rozdělení jsem přistoupil na základě rozdílných funkcionalit a pro větší přehlednost.

### 4.1 Formát vstupních dat

Vstupní data jsou zakódována do běžně používaného formátu CSV, v modifikaci používané v České republice na platformě MS-Windows. Tj. data jsou oddělena místo čárkou středníkem, jako desetinný oddělovač je použita čárka a řádky jsou odděleny kombinací znaků \CR a \LF. Kódování souboru je nastaveno na znakovou sadu Windows CP1250, avšak protože soubor obsahuje pouze čísla, středník, čárka a řídicí znaky, je kompatibilní s kódováním ASCII a tím i se standardním UTF-8 (dle [10]).

ASCII Code Chart																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Obrázek 4.1: Společná část kódování ASCII, CP1250 a UTF-8 (zdroj:[1])

Data jsou rozdělena do dvou sloupců, první obsahuje souřadnice na ose X, druhý souřadnice na ose Y. Každá ze souřadnic je zapsaná ve formě celé části čísla a třech

desetinných míst oddělených čárkou (jako text).

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	362C	3134	333B	352C	3836	350D	0A36	2C31	6	,	1	4	3	;	5	,	8	6	5	.	.	6	,	1
0010	3036	3B35	2C38	3238	0D0A	362C	3034	383B	0	6	;	5	,	8	2	8	.	.	6	,	0	4	8	;
0020	352C	3831	300D	0A36	2C30	3336	3B35	2C37	5	,	8	1	0	.	.	6	,	0	3	6	;	5	,	7
0030	3836	0D0A	362C	3030	303B	352C	3737	390D	8	6	.	.	6	,	0	0	0	;	5	,	7	7	9	.
0040	0A35	2C39	3438	3B35	2C37	3732	0D0A	352C	5	9	4	8	.	5	7	7	2	5						

Obrázek 4.2: Ukázka formátování dat v souboru křivky (zvýrazněn jeden řádek)

Program předpokládá existenci dvou takovýchto souborů (dvou křivek k vzájemnému porovnání). Soubory musí být ve stejném adresáři a jména souborů se musí lišit pouze v koncové části. Jeden soubor musí končit řetězcem „\_R.csv“ a druhý řetězcem „\_L.csv“. Tímto jsou od sebe odlišeny strany jednotlivé křivky podle stran, při otevírání jednoho ze souborů je druhý dohledán automaticky (pokud chybí, je program zastaven).

## 4.2 Grafické transformace

Po načtení obou křivek je zahájena smyčka, ve které může uživatel upravovat vzájemné posunutí a natočení levé křivky vůči pravé. Důvodem proč toto není implementováno i naopak je ten, že jde o porovnání dvou křivek a tedy jde o relativní vzdálenosti a ne o vzdálenosti vůči souřadnému systému.

Tedy jakákoliv grafická transformace provedená na jedné ze křivek bude mít stejné následky jako tatáž transformace provedená na druhé ze křivek, avšak s opačnými parametry.

### 4.2.1 Translace

První použitou grafickou transformací je translace neboli posunutí. To je možné realizovat dvěma způsoby. Prvním z nich je za pomoci translační transformační matice 4.1. Touto maticí se vynásobí vektor starých souřadnic rozšířený o jedničku a výsledkem je vektor souřadnic (opět rozšířený o jedničku) posunutých o vektor umístěný v třetím sloupci transformační matice.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.1)$$

Druhým ekvivalentním způsobem je prosté přičtení hodnoty posunu ke zvolené souřadnici. Ekvivalence obou zápisů může být dokázána roznásobením matice a vektoru 4.2.

$$\begin{aligned} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &\downarrow \\ x' &= 1 \cdot x + 0 \cdot y + s_x \cdot 1 \\ y' &= 0 \cdot x + 1 \cdot y + s_y \cdot 1 \\ 1 &= 0 \cdot x + 0 \cdot y + 1 \cdot 1 \\ &\downarrow \\ x' &= x + s_x \\ y' &= y + s_y \end{aligned} \quad (4.2)$$

Ve výše uvedených zápisech  $s$  reprezentuje vektor posunutí,  $x$  a  $y$  reprezentují původní souřadnice bodu a  $x'$  a  $y'$  reprezentují nové souřadnice. V programu byl použit druhý způsob a to z důvodu posunu vždy jen jedním směrem v daný okamžik.

## 4.2.2 Rotace

Další z použitých grafických transformací je rotace čili natočení. Opět se jedná o transformaci, která se obvykle provádí pomocí transformační matice. Tentokrát však není úplně jednoznačná, závisí na tom, zda je úhel natočení pozitivní ve směru hodinových ručiček, či protisměru. Podle toho se mění znaménko u členů se sinem.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.3)$$

V případě 4.3 se jedná o matici rotace, kde pozitivní hodnota parametru  $\varphi$  způsobuje rotaci po směru hodinových ručiček. Násobením uvedené matice a vektoru opět



můžeme získat soustavu rovnic pro nové souřadnice (dle 4.4).

$$\begin{aligned}
 \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
 &\downarrow \\
 \begin{aligned} x' &= \cos(\varphi) \cdot x + \sin(\varphi) \cdot y + 0 \cdot 1 \\ y' &= -\sin(\varphi) \cdot x + \cos(\varphi) \cdot y + 0 \cdot 1 \\ 1 &= 0 \cdot x + 0 \cdot y + 1 \cdot 1 \end{aligned} & \quad (4.4) \\
 &\downarrow \\
 \begin{aligned} x' &= \cos(\varphi) \cdot x + \sin(\varphi) \cdot y \\ y' &= -\sin(\varphi) \cdot x + \cos(\varphi) \cdot y \end{aligned}
 \end{aligned}$$

I v tomto případě bylo použito transformace pomocí složkového tvaru, ale zde již bylo možné bez větších problémů použít i maticový tvar výpočtu.

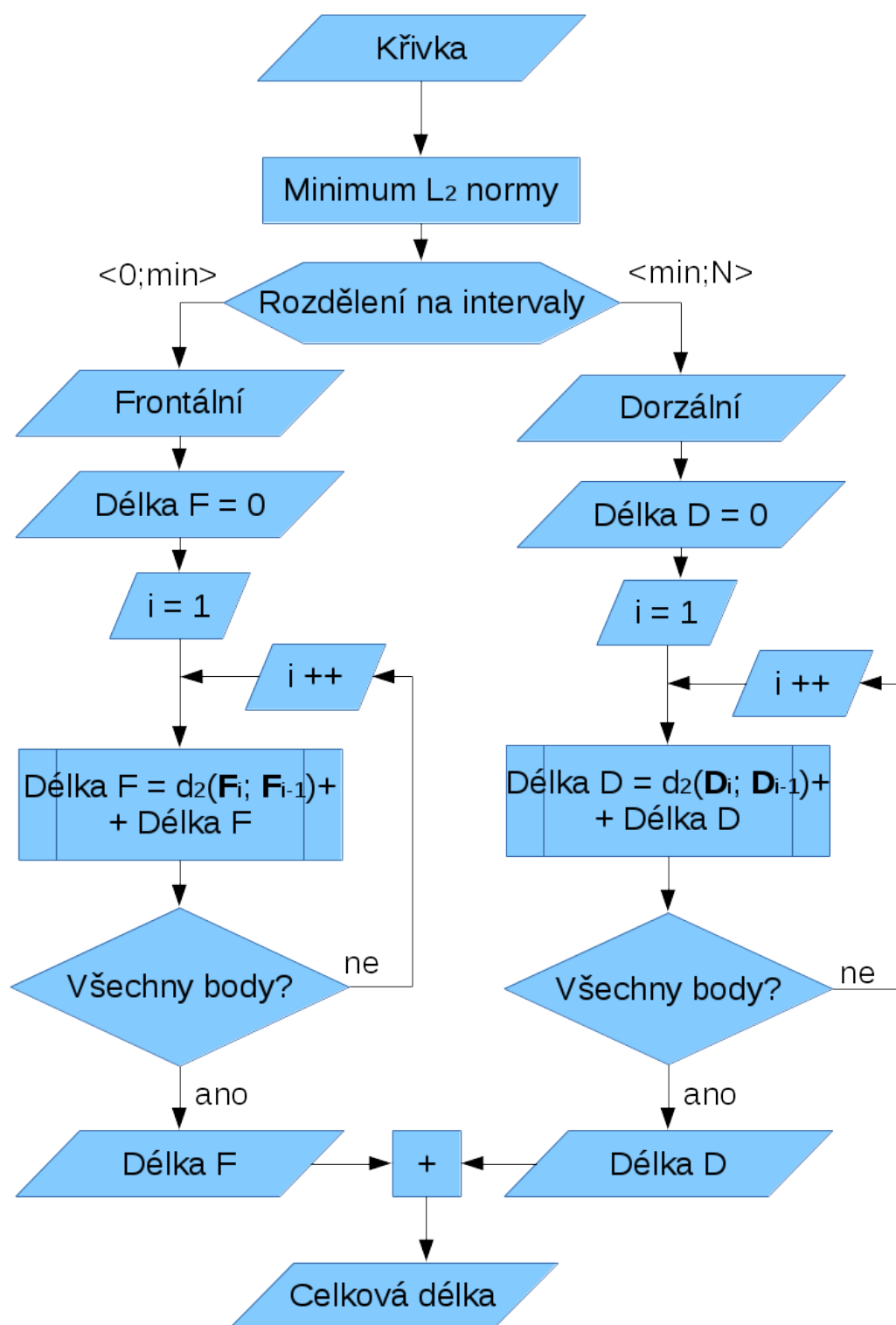
### 4.3 Analýza délky křivky

Při analýze délky křivky bylo nutné provést několik aproximací. Je totiž nutné uvědomit si, že data byla získána převodem z prostoru  $\mathbb{N}^2$  do prostoru  $\mathbb{R}^2$ . Kde prostor  $\mathbb{N}^2$  je nespojitý prostor celých kladných dvojic čísel tvořící vektory. Tento prostor je klasickou reprezentací obrazových dat ve formě bitmap rastrové grafiky. Neboli každá souřadnice pixelu je celým číslem. Jak bylo již popsáno dříve, přepočtení dat uskutečněno násobením odečteného měřítka původně celočíselnými souřadnicemi s posunutým středem soustavy do středu acetabula. Při tomto přepočtu však zvětšují rozdíly mezi jednotlivými body, takže vzniká linearizovaná spojitá po částech diferencovatelná křivka (s nespojitou první derivací).

Úlohu výpočtu délky křivky pak lze realizovat za pomoci  $L_2$  metriky (Eukleidovské metriky) pro měření vzdálenosti krajních bodů takto vzniklých úseček. Suma těchto dílčích metrik pak odpovídá délce křivky.

$$d_p(a; b) = \left( \sum_{i=1}^n |a_i - b_i|^p \right)^{\frac{1}{p}}, p \in \mathbb{N}, n \in \mathbb{N} \quad (4.5)$$

Rovnice 4.5 vyjadřuje obecnou definici  $L_p$  metriky ze které lze získat požadovanou



Obrázek 4.3: Algoritmus pro výpočet délky křivky

$L_2$  metriku na prostoru  $\mathbb{R}^2$  pomocí dosazení dvojky jak za parametr  $p$ , tak za parametr  $n$ . Parametr  $p$  zde reprezentuje stupeň metriky, parametr  $n$  dimenzi prostoru. Vstupními hodnotami  $a, b$  se rozumí body v prostoru s danou dimenzí  $n$ .

$$\begin{aligned} d_2(a; b) &= \left( \sum_{i=1}^2 |a_i - b_i|^2 \right)^{\frac{1}{2}} \\ l_c &= \sum_{i=1}^N d_2(X_{i-1}; X_i), X = (x_i; y_i) \\ &\quad \downarrow \\ l_c &= \sum_{i=1}^N \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2} \end{aligned} \tag{4.6}$$

Délka křivky je počítána samostatně na obou křivkách ve dvou intervalech. To vyplývá z požadavků vedoucího, aby bylo možné stanovit požadovanou délku vnitřní pánevní dlahy jak frontálně, tak dorzálně od acetabula. Takto získaná data pak mohou být použita například pro porovnání parametrů levé a pravé křivky, ověření hypotéz existence univerzálních délek nebo případné korelace délky dle skupin pacientů (například podle pohlaví).

## 4.4 Analýza křivosti

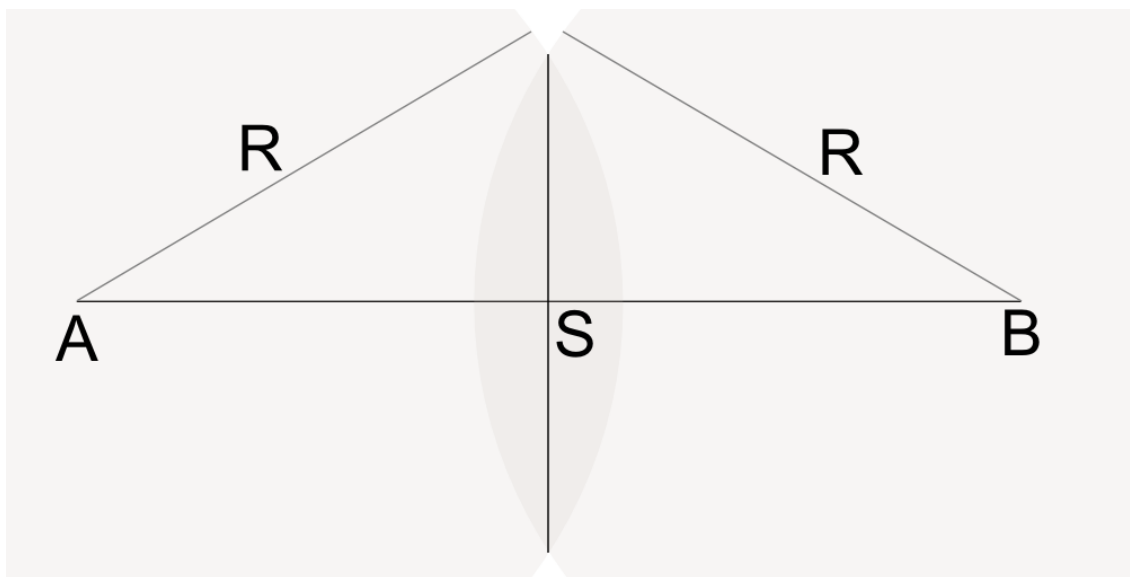
Pro získání údajů o poloměru křivky je nejprve zapotřebí seznámit se z některými pojmy „pokročilé“ matematiky. V první řadě je nutná znalost matematických prostorů, hlavně jde o prostory normované, což je nutná podmínka pro výpočty, které jsou uvedeny dále. Další znalostí potřebnou pro úplné porozumění je teorie kolem metrik, či alespoň Eukleidovské metriky, která je v programu použita.

### 4.4.1 Původní záměr

Původním záměrem bylo radius křivky měřit pomocí několika náhodně vybraných bodů (minimálně tří), ze kterých by bylo možné najít střed prokládané kružnice. Tento střed kružnice lze ve zkratce najít jako bod ve kterém  $L_2$  metriky ke všem zvoleným bodům rovnají. Poloměr prokládané kružnice se pak rovná této společné hodnotě  $L_2$  metriky.

Geometricky si lze představit tuto úlohu obdobně jako hledání středu úsečky pomocí kružítka. Tedy do kružítka se nastaví vzdálenost delší než je hledaná polovina úsečky. Poté se vynesou relevantní části kružnic z obou krajních bodů. Tyto části kružnic

se protnou přesně ve dvou bodech a jejich propojením vznikne průsečík s úsečkou, který odpovídá jejímu středu.



Obrázek 4.4: Nalezení středu úsečky pomocí kružítka

Podobně lze postupovat i v tomto případě, avšak zde projetím průsečíků kružnic z krajních bodů nezískáme přímo polohu středu prokládané kružnice, ale pouze přímku na které se nachází. Další úvahou je možné tuto přímku redukovat na polopřímku. To lze docílit porovnáním křivky s úsečkou propojující dva krajní zvolené body. Na té straně úsečky, kde se nalézá třetí zvolený bod se od křivky dál nemůže nacházet střed kružnice. Průsečík původní přímky určující polohu středu a křivky je hraničním bodem vzniklé polopřímky. Pokud by došlo k případu, kdy zvolené tři body leží na přímce, tak tuto metodu není možné použít, protože by střed kružnice vycházel v nekonečno na obou stranách přímky.

Když budeme počítat, že v prvním kroku nenastaly žádné potíže, tak lze přistoupit k samotnému výpočtu vzdálenosti. To lze realizovat jako úlohu hledání vázaného extrému funkce rozdílu metrik. Příkladem minimalizované funkce může být například tato funkce 4.7, kde  $a$  a  $b$  jsou krajní body,  $c$  je prostřední bod a  $S$  odpovídá hledanému středu kružnice.

$$d_2(a; S) + d_2(b; S) - 2d_2(c; S) = 0 \quad (4.7)$$

Z uvedeného postupu již vyplývá jeho hlavní nevýhoda. Ta spočívá v silné závislosti na zvolených bodech. Jelikož získaná křivka rozhraní kost–ostatní tkán nepřípominá

ani zdaleka kružnici, nelze postupovat pomocí náhodného výběru bodů, aniž by se to podepsalo významnou měrou na přesnosti naměřených dat. Příkladem nevhodné křivky pro tuto metodu může být křivka, která má více úseků s rozdílným poloměrem.

#### 4.4.2 Minimalizace funkce metodou nejmenších čtverců

Jelikož výše uvedený příklad více méně odpovídá řešenému problému, tak bylo nutné přijít s řešením, které nezáleží na pouze omezené podmnožině získaných bodů na zkoumaném intervalu. Místo toho bylo nutné použít všechny body. K tomuto účelu se náramně hodí minimalizace funkce metrik mezi uvažovanou částí kružnice a body křivky. K samotnému řešení parametrů je zde pak použita metoda nejmenších čtverců s aproximací kružnicí.

$$(x - m)^2 + (y - n)^2 = r^2 \quad (4.8)$$

Na začátku vyháším z obecné rovnice kružnice 4.8, kde  $x$  a  $y$  jsou body kružnice,  $m$  je Xovou souřadnicí středu,  $n$  je Yovou souřadnicí středu a  $r$  je poloměrem kružnice. Z této rovnice je pak dále tvořena funkce, jejíž suma pro všechny body křivky je minimalizována. Pro tento účel musí být upravena do tvaru 4.9, kde zůstávají jako parametry  $m$ ,  $n$  a  $r$ . Písmena  $x$  a  $y$  jsou zde ale body křivky a ne kružnice, jako tomu bylo v rovnici 4.8.  $N$  odpovídá počtu bodů křivky ve vyšetřovaném rozdhahu.

$$F(m, n, r) = \sum_{i=1}^N [(x_i - m)^2 + (y_i - n)^2 - r^2]^2 \quad (4.9)$$

$$\begin{aligned} \frac{\partial F}{\partial m} &= \sum 2(m - x_i) = 0 \\ \frac{\partial F}{\partial n} &= \sum 2(n - y_i) = 0 \\ \frac{\partial F}{\partial r} &= -2r = 0 \end{aligned} \quad (4.10)$$

Klasickým řešením je položení parciálních derivací, podle všech proměnných, rovno nule a řešit systém rovnic o více neznámých, jak to ukazují rovnice 4.10. Další možností je linearizace této funkce a její převedení do maticového tvaru tak zvané soustavy normálních rovnic. Vzhledem k většímu počtu dat byla zvolena druhá metoda řešení. Postup linearizace ukazují rovnice 4.11. Průběh úpravy spočívá nejprve v roznásobení závorek, poté převedení všech členů krom  $x^2$  a  $y^2$  na pravou stranu.

Dalším krokem je substituce 4.12, čímž je docíleno lineární závislosti rovnice na zkoumaných parametrech.

$$\begin{aligned}
(x-m)^2 + (y-n)^2 &= r^2 \\
\downarrow \\
x^2 - 2mx + m^2 + y^2 - 2ny + n^2 &= r^2 \\
\downarrow \\
x^2 + y^2 &= 2mx + 2ny - m^2 - n^2 + r^2 \\
\downarrow \\
x^2 + y^2 &= Ax + By + C
\end{aligned} \tag{4.11}$$

$$\begin{aligned}
A &= 2m \\
B &= 2n \\
C &= r^2 - m^2 - n^2
\end{aligned} \tag{4.12}$$

Metoda soustavy normálních rovnic 4.13 je založena na rozšíření původní soustavy rovnic  $\mathbf{Ax} = \mathbf{b}$  na obou stranách rovnice transponovanou maticí  $\mathbf{A}^\top$ . Tím se z původně obdélníkové matice typu  $\mathbb{R}^{m \times n}$ , kde  $m > n$ , stane čtvercová matice typu  $\mathbb{R}^{n \times n}$ . Je však nutné si uvědomit, že se nejedná o soustavu ekvivalentní, tedy že pokud má původní soustava řešení, tak řešení normální soustavy je řešením soustavy původní. Avšak již nemusí platit, že řešení původní soustavy je zároveň i řešením soustavy normální. Původní soustava je totiž zpravidla přeurčená, zatímco normální soustava má stejný počet rovnic jako parametrů.

$$\mathbf{Ax} = \mathbf{b} \rightarrow \mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b} \tag{4.13}$$

Výpočet matice normální soustavy je vidět v rovnici 4.14, kde  $x_1$  až  $x_n$  a  $y_1$  až  $y_n$  odpovídají dvojicím souřadnic dat a  $n$  zde znamená počet dat. Jak je v rovnici vidět, tak původní data jsou pro další výpočty nahrazena jejich součty, součty kvadrátů nebo součty součinů. Hlavní výhodou je však to, že je výsledná matice čtvercová a tedy lze použít většinu klasických metod lineární algebry (kupříkladu je možné při splnění dalších kritérií vytvořit inverzní matici, místo matice pseudoinverzní u obdélníkových matic).

$$\begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \quad (4.14)$$

$$\begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix} = \begin{bmatrix} \sum x_i (x_i^2 + y_i^2) \\ \sum y_i (x_i^2 + y_i^2) \\ \sum (x_i^2 + y_i^2) \end{bmatrix} \quad (4.15)$$

Obdobně probíhá výpočet i u pravé strany (4.15), kde je vynásobením pravé strany linearizované funkce transponovanými daty získána pravá strana normální soustavy. Po jejím vyjádření je zkompletována normální soustava 4.16, která je následně předána některému řešiči soustav lineárních rovnic. Bez bližšího určení vlastností matice se nabízí z iteračních metod například metoda minimálních zobecněných residuí (GMRES), u které stačí aby byla matice soustavy regulární.

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x_i (x_i^2 + y_i^2) \\ \sum y_i (x_i^2 + y_i^2) \\ \sum x_i^2 + y_i^2 \end{bmatrix} \quad (4.16)$$

Po vyřešení normální soustavy 4.16 se lze k původním hledaným parametrům  $m$ ,  $n$  a  $r$  dostat pomocí zpětné substituce 4.17. Výsledkem čehož získáme jak souřadnice středu kružnice, tak její poloměr, který může být využit k popisu křivky.

$$\begin{aligned} m &= \frac{A}{2} \\ n &= \frac{B}{2} \\ r &= \frac{\sqrt{4C + A^2 + B^2}}{2} \end{aligned} \quad (4.17)$$

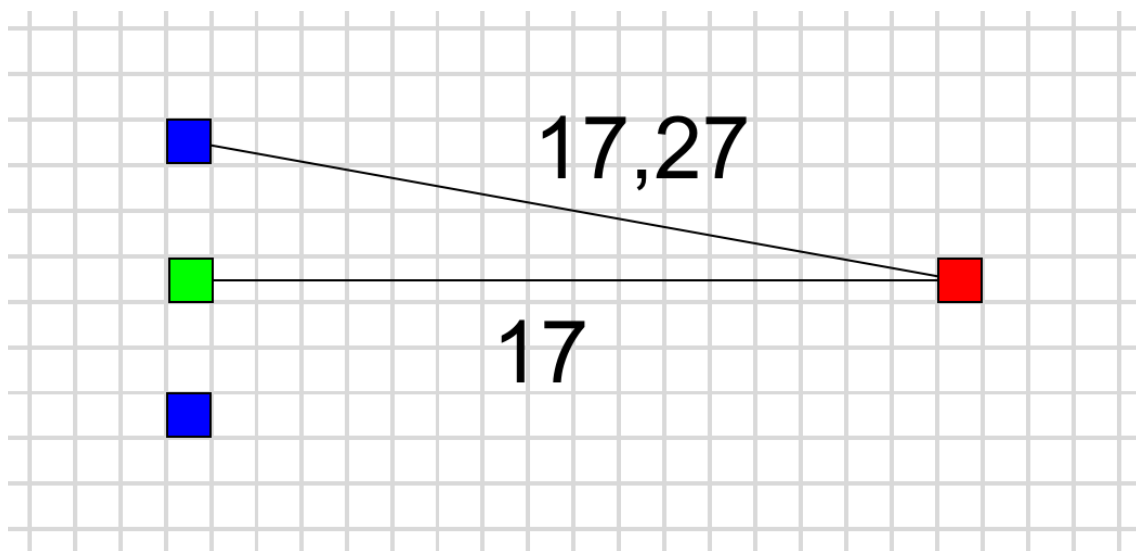
V této kapitole byla problematika prokládání kružnice křivkou zmíněna jen povrchně a v dosti zjednodušené formě. Případné zájemce o podrobnější informace bych odkázal na zdroj [3] (konkrétně kapitolu 6), případně na zdroj [9] odkud jsem čerpal hlavně v počátečních fázích vývoje.

## 4.5 Analýza vzdálenosti křivek

Pro analýzu vzdálenosti křivek je použito opět Eukleidovské metriky, nebo přesněji součet čtverců těchto metrik. Samotná metrika se zde počítá z aproximace vzdálenosti libovolného bodu od nejbližšího bodu druhé křivky. Aproximace spočívá ve skutečnosti, že místo toho, aby se vzdálenosti počítaly ze vzdálenosti bodu a spojnice bodů druhé přímky, zde bylo použito minimální vzdálenosti bodů obou přímek.

$$\lim_{d_2(a;b) \rightarrow \infty} d_2(a; b) - d_2(a'; b) = 0, d_2(a - a') \ll d_2(a; b) \quad (4.18)$$

Z rovnice 4.18 vyplývá, že při použití této aproximace nedochází k výrazným chybám pokud je vzdálenost křivek výrazně větší než vzdálenost sousedních bodů křivky. V uvedené rovnici body  $a$  a  $a'$  odpovídají jedné ze křivek a bod  $b$  druhé. Bod  $a'$  je aproximací bodu s minimální vzdáleností ke zkoumanému bodu  $b$ . Tento závěr lze vyvodit i z obrázku 4.5, kde modré body odpovídají jedné křivce, červený bod druhé a zelený bod je interpolovaný nejbližší bod ke druhé křivce. Tento obrázek znázorňuje nejhorší možný případ, kdy interpolovaný bod vychází přesně doprostřed intervalu mezi body  $a$  a zároveň jsou body křivky od sebe poměrně dost vzdáleny. I tak zde dochází k chybě přibližně 1,6 %. V praxi je tato chyba kvůli malým vzdálenostem mezi body v křivce ještě menší.



Obrázek 4.5: Vliv aproximace na výslednou vzdálenost

Rovnice pro výpočet průměrné vzdálenosti bodů křivek je uvedena pod číslem 4.19.



Přičemž  $a_i$  odpovídá souřadnicím bodů první křivky,  $b_j$  bodům druhé křivky,  $M$  je počet bodů první křivky,  $N$  je počet bodů druhé křivky a  $\bar{v}$  je ona průměrná vzdálenost. Ve zkratce jde o to, že při hledání průměrné vzdálenosti jsou postupně procházeny všechny body křivky a k nim jsou vyhledávány nejbližší body. Vzdálenosti bodů jsou v první mocnině sčítány a výsledek je vydělen počtem sčítanců.

$$\bar{v} = \frac{1}{M} \sum_{i=1}^M \min_{j \in \{1;N\} \cap \mathbb{N}} d_2(a_i; b_j) \quad (4.19)$$

Podobně lze postupovat i v případě nalezení maximální vzdálenosti, která odpovídá maximálnímu nutnému ohnutí dlahy pokud by byla modelována podle opačné strany. V rovnici 4.20 je vidět matematické vyjádření této úlohy. Zde bych jen upozornil, že zde již nejsou sčítány  $d_2$  metriky, ale pouze hledána maximální vzdálenost mezi dvěma nejbližšími body.

$$v_{max} = \max_{i \in \{1;M\} \cap \mathbb{N}} \left[ \min_{j \in \{1;N\} \cap \mathbb{N}} d_2(a_i; b_j) \right] \quad (4.20)$$

Vzdálenost křivek je v programu z několika důvodů. Prvním z nich je, že poskytuje údaje o přesnosti sesazení levé křivky na pravou. To slouží k porovnání vizuální podobnosti obou křivek a určitému prvotnímu zhodnocení, zda se u daného řezu vůbec o porovnání stran snažit (například kvůli zlomeninám nebo kvůli špatným datům). Jako druhý důvod lze označit výpočet průměrné vzdálenosti obou křivek a tedy určitou míru shody, kterou lze později použít spolu s ostatními získanými parametry pro vyhodnocení hypotézy o shodě obou stran. Třetím důvodem je již zmíněné získání informace o maximálním ohnutí dlahy modelované podle druhé strany pánve.

## 4.6 Implementace programu

Program je rozdělen sekvenčně do sedmy kroků a obsahuje jednu doprogramovanou funkci. Tato funkce řeší proložení kružnice křivkou a tím měření poloměrů v místech zakřivení dlahy. Funkci jsem pracovně nazval „Circular fit“ a její druhou revizi můžete vidět na obrázku 4.6. Druhá revize má oproti původní zcela odlišnou vnitřní strukturu s použitím maticových variant běžných funkcí. Zároveň zde byla opravena chyba ve zpětné substituci jak jí uváděl zdroj [9]. V rovnici 4.17 je chyba opravena.

### 4.6.1 Funkce Circular fit

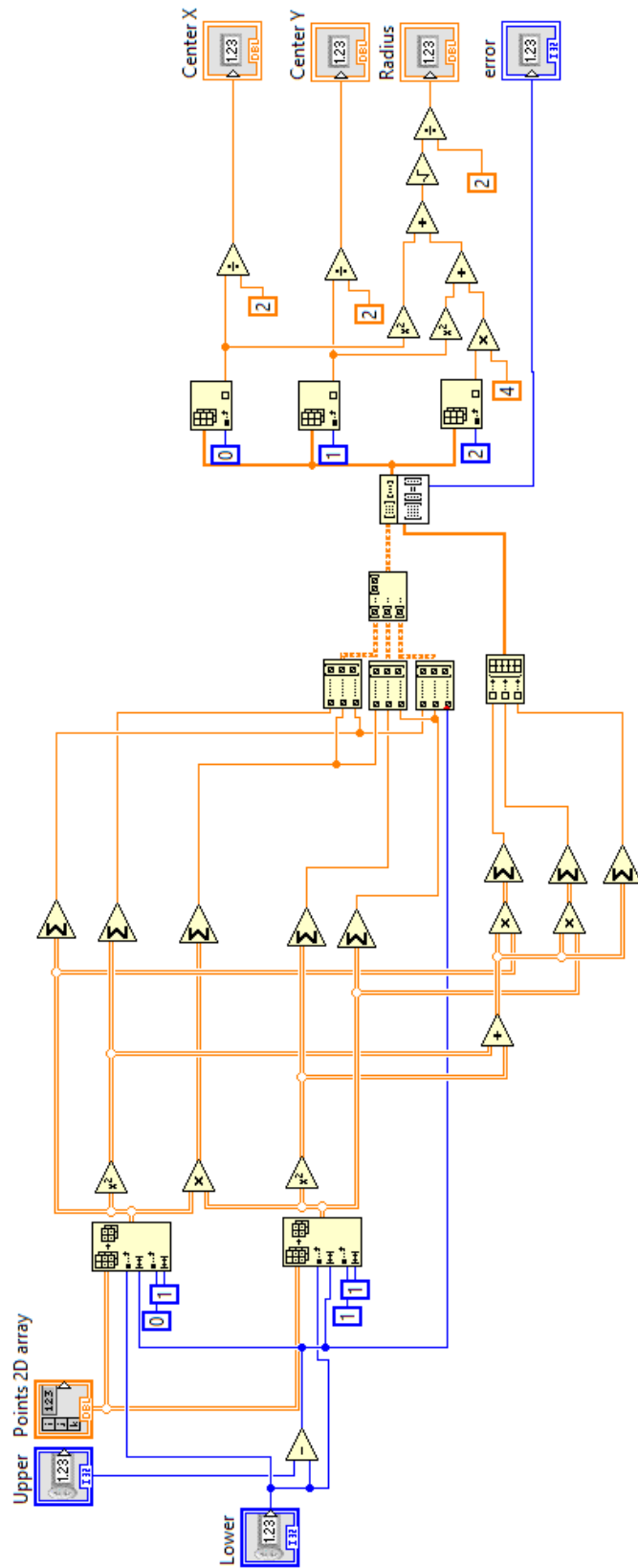
Do funkce vstupuje dvourozměrné pole souřadnic bodů křivky. Z něj jsou vytvořeny dvě podmnožiny. První je tvořena souřadnicemi na ose  $X$  v intervalu specifikovaným uživatelem, druhá je tvořena souřadnicemi  $Y$  na stejném rozsahu. Rozsah ve kterém se prokládá křivka kružnicí je určen dvěma vstupy „Upper“ a „Lower“. Oba tyto vstupy předpokládají celočíselnou hodnotu a odpovídají indexům v poli souřadnic. Měly by se tedy nacházet u intervalu od nuly do délky pole a zároveň v oboru celých kladných čísel včetně nuly.

Z těchto dvou podmnožin je následně vytvořena matice soustavy a vektor pravé strany jak je uvedeno v rovnici 4.16. Takto vzniklý systém rovnic je předán výchozímu řešiči v prostředí LabView. Přičemž jako parametr je zde specifikována nejobecnější možná matice pouze s předpokladem regularity. To by mělo u rozměrnějších matic za následek větší náročnost algoritmu na paměť a výpočetní čas, ale tím že byla matice soustavy pomocí metady normálních rovnic redukována na rozměr  $3 \times 3$ , tak je paměťová náročnost metody GMRES zanedbatelná.

Po vyřešení soustavy je provedena zpětná substituce a tedy výpočet požadovaných parametrů souřadnic středu a poloměru kružnice. Dále funkce předává chybu iteračního řešiče a tím signalizuje chyby vstupních dat. Pokud by například měla soustava singulární matici, tak nebude řešitelná (vznikne parametrické řešení) a reziduum bude po zaokrouhlení nenulové.

### 4.6.2 Hlavní program – příprava a zobrazení dat

Program samotný začíná smyčkou čtení tlačítka pro načtení hodnot. Po jeho stisku program přechází do dalšího kroku, kdy je otevřen souborový dialog. V tomto dialogu je uživatel vyzván k otevření souboru dat s příponou CSV. Je jedno jakou ze stran uživatel zvolí, vždy se otevřou obě. Pokud jeden ze souborů neexistuje je program automaticky zastaven. Nebyla by totiž splněna nutná podmínka pro srovnání obou stran. V případě, že by uživatel chtěl pouze vyhodnocovat nějakou obecnou křivku musí krom těchto dat vytvořit ještě jeden soubor s názvem ve formátu „NázevDat\_Strana.csv“ kde *strana* musí být „R“ nebo „L“, ale musí být zastoupeny obě varianty. Tedy například má data křivky se budou jmenovat „data.csv“ a budou splňovat požadavky z kapitoly Formát vstupních dat. Tak pro zpracování dat musím soubor s daty přejmenovat na „data.L.csv“ (posouvání a otáčení je prováděno pouze



Obrázek 4.6: Druhá revize funkce Circular fit

na levé křivce). Dále musím vytvořit soubor s názvem „data\_R.csv“ a doporučuji do něj zapsat alespoň jeden bod například „0,000;0,000“ a ten odřádkovat. Tím by měly jakákoliv data projít kontrolou ve čtvrtém kroku programu.

Třetí krok vytváří z cesty k jednomu ze souborů dočasnou cestu z níž je pak ve čtvrtém kroku vytvářena cesta k souborům obou stran. Tím je zajištěna nezávislost na tom jaký ze souborů uživatel zvolil v souborovém dialogu. Zároveň je ve čtvrtém kroku provedena i kontrola existence obou souborů, jak bylo uvedeno výše.

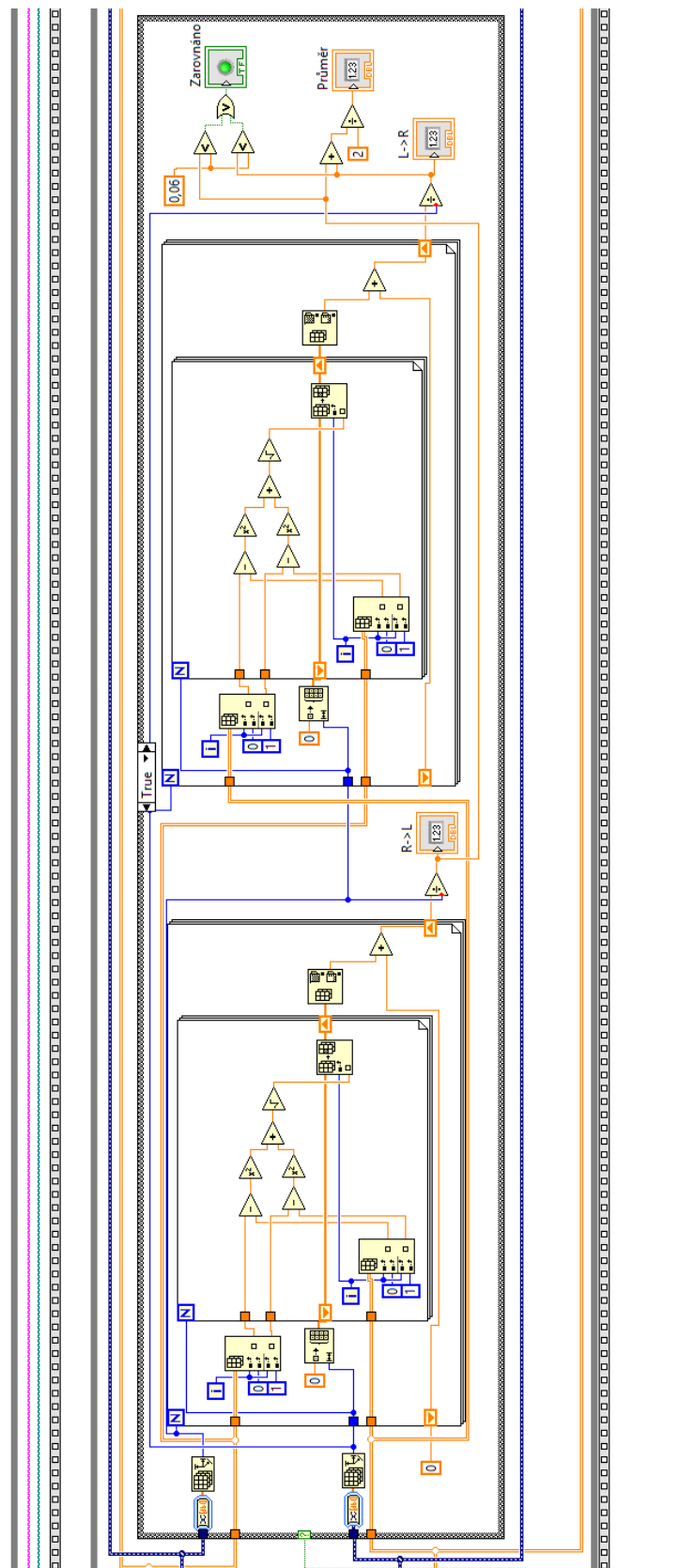
Pátým krokem je načtení dat z obou souborů a zobrazení pracovní cesty. Jak již bylo popsáno dříve v kapitole Formát vstupních dat, tak jako řádkový oddělovač jsou použity znaky `\CR` a `\LF` jako je v prostředí Microsoft Windows běžné. Jako sloupcový oddělovač je použit středník a jako desetinný oddělovač čárka. Výstupní datový typ je `double`, tedy 64b datový formát s takzvanou dvojitou přesností. Výsledkem je tedy pole o dvou sloupcích (souřadnice  $X$  a  $Y$ ) a  $N$  řádcích, kde  $N$  odpovídá počtu bodů křivky.

Šestý krok je hlavní početní částí programu. V části z této podkapitoly jsou prováděny grafické transformace posuvu a rotace a následné zobrazení obou křivek uživateli. K transformacím dochází vždy při stisku příslušných tlačítek, kdy se do dráhy souřadnic levé strany vřadí příslušný blok. Navíc dojde k přičtení jedničky do proměnné, která se v každém cyklu nuluje a její nenulová hodnota signalizuje, že je nutné přepočítat hodnoty měření vzdálenosti. Zdrojový kód této podkapitoly je uveden na obrázku 4.7.

### 4.6.3 Měření vzdálenosti křivek

Měření vzdálenosti křivek probíhá ve dvou krocích. Nejdříve jsou procházeny body jedné ze křivek a k něm jsou hledány body s nejmenší vzdáleností a následně se tento postup opakuje pro druhou ze křivek. Hlavním důvodem je typicky různá délka křivek, která pak vede k diametrálně odlišným hodnotám vypočítaných parametrů. Průměr těchto dvou hodnot se ukázal jako nevhodný parametr, protože je na různou délku podobně citlivý jako jednotlivé hodnoty. Naopak hodnocení podle menších z obou hodnot se ukázalo být více průkazné. Hodnota 0,06 cm jak je vidět na obrázku 4.8 je však dosažitelná pouze u křivek s minimálními odchylkami. U křivek s hodnotou do 0,15 cm je ještě stále dosti patrná podobnost.





Obrázek 4.8: Měření průměrné vzdálenosti mezi křivkami

Tyto výpočty se provádějí pokaždé, když je provedena jakákoliv geometrická transformace. Tím je zajištěno, že je uživatel informován o výsledcích svého snažení a podle toho měnit postup sesazování jedné křivky na druhou. Zároveň však v případě, že k žádné změně nedochází, tak jsou šetřeny systémové prostředky pro další procesy, tento blok se potom nevykonává.

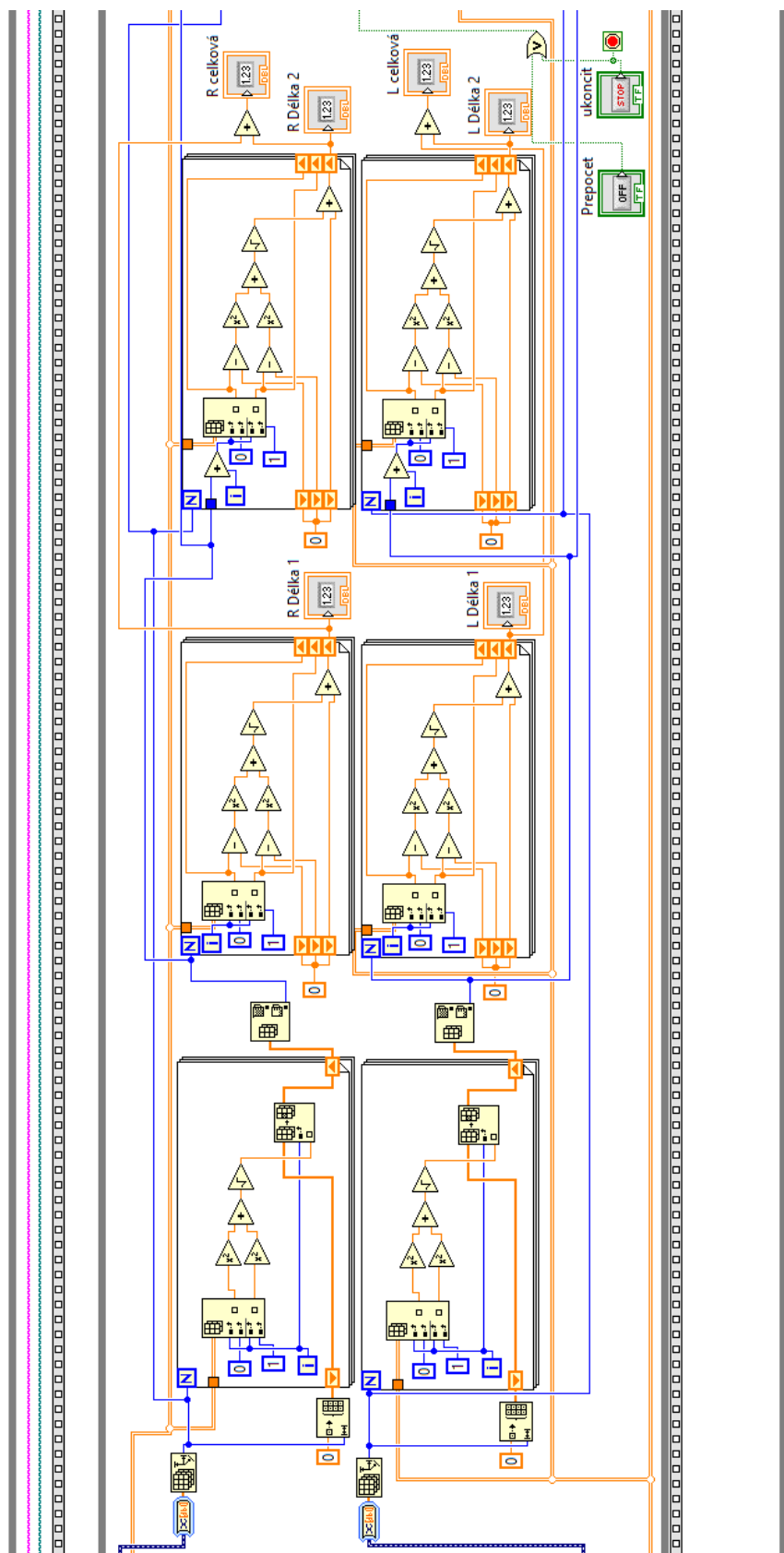
#### 4.6.4 Měření délek

Před měřením délek je nejdříve nutné najít bod s nejmenší vzdáleností od středu soustavy. Tento bod odpovídá bodu s nejkratší vzdáleností od středu acetabula. Po jeho nalezení je možné podle jeho indexu rozdělit křivku na dvě části. Část frontální a část dorzální. Přičemž platí, že body v nižším indexem než je minimum odpovídají frontální části a s větším dorzální. A ty poté dále vyhodnocovat samostatně.

Po nalezení indexu minima je možné začít vyhodnocovat délku jednotlivých segmentů. To je zajištěno pomocí postupného sčítání Eukleidovské vzdálenosti mezi sousedními body. Takto jsou spočítány postupně všechny čtyři délky (dvě pro každou ze křivek), načež po sečtení přední a zadní části křivky je získána celková délky křivky. Programový kód je na obrázku 4.9 spolu s podmínkou pro provedení výpočtu proložené křivky.

#### 4.6.5 Proložení kružnice a uložení výsledků

Výpočet prokládané kružnice se provádí pouze pokud k tomu dá uživatel příkaz. Může tak učinit buďto pomocí tlačítka „Přepočítat“, nebo tlačítkem „Konec úprav“, které navíc způsobí, že program přejde do fáze ukládání dat. Důvodem toho, že je tento výpočet prováděn pouze na vyžádání je jeho poměrně velká náročnost na systémové prostředky. Výpočet se totiž skládá z řešení čtyř lineárních soustav včetně přípravy matic a vektorů pravých stran doplněný o přípravu intervalů. Pokud je tento postup opakován při každém pohybu křivky, tak doba potřebná pro vykonání pohybu se prodlouží za hranici jedné sekundy, čímž se aplikace stává z uživatelského hlediska nepoužitelnou. Zatím co když je přepočet na vyžádání a tlačítko pro jeho spuštění je navíc na druhé straně než je zbytek uživatelského rozhraní, tak uživatel se nestačí ani vrátit myši k ovládacím prvkům pro pohyb a výpočet je dokončen. Aplikace se tak na oko jeví rychlejší.



Obrázek 4.9: Hledání bodu s minimální vzdáleností od středu a měření délek



Interval na kterém je prokládání realizováno může uživatel ovlivnit pomocí dvou dvojitych posuvníků. Takže pokud uživatel vidí, že je křivka zahnutá jen na určitém intervalu a jinak je téměř rovná, může vybrat jen onu prohnutou oblast a získá poloměr onoho prohnutí. Indexy hranic intervalů lze získat vydělením stem (posuvník nabývá hodnot nula až sto), vynásobením počtem hodnot v původním intervalu a zaokrouhlením. V případě, že se jedná o dorzální část křivky je ještě nutné přičíst hodnotu indexu středu acetabula.

V momentě kdy jsou vypočtené hranice intervalů jsou data předána čtyřem instancím funkce „Circular fit“. Ty zajistí výpočet poloměrů jak je blíže popsáno v podkapitolách Minimalizace funkce metodou nejmenších čtverců a Funkce Circular fit.

Pokud je uživatel spokojen se zarovnáním křivek a vypočtenými daty může pokračovat pomocí tlačítka „Konec úprav“ do posledního kroku programu, ukládání dat. Nejdříve je uživatel dotázán zda chce získaná data uložit. Pokud odpoví, že ne, program dokončí cyklus a umožní načtení nových křivek. Pokud odpoví, že si přeje data uložit, program otevře souborový dialog pro výběr souboru s výsledky. Výchozí název souboru je „results.csv“ a výchozí umístění je složka se vstupními daty. Když uživatel zvolí neexistující soubor, je tento soubor vytvořen včetně hlavičky s názvy parametrů a je do něj přidán záznam aktuálních dat. Pokud již soubor existuje jsou data přidána do nového řádku na konec souboru.



## 4.7 Uživatelské rozhraní

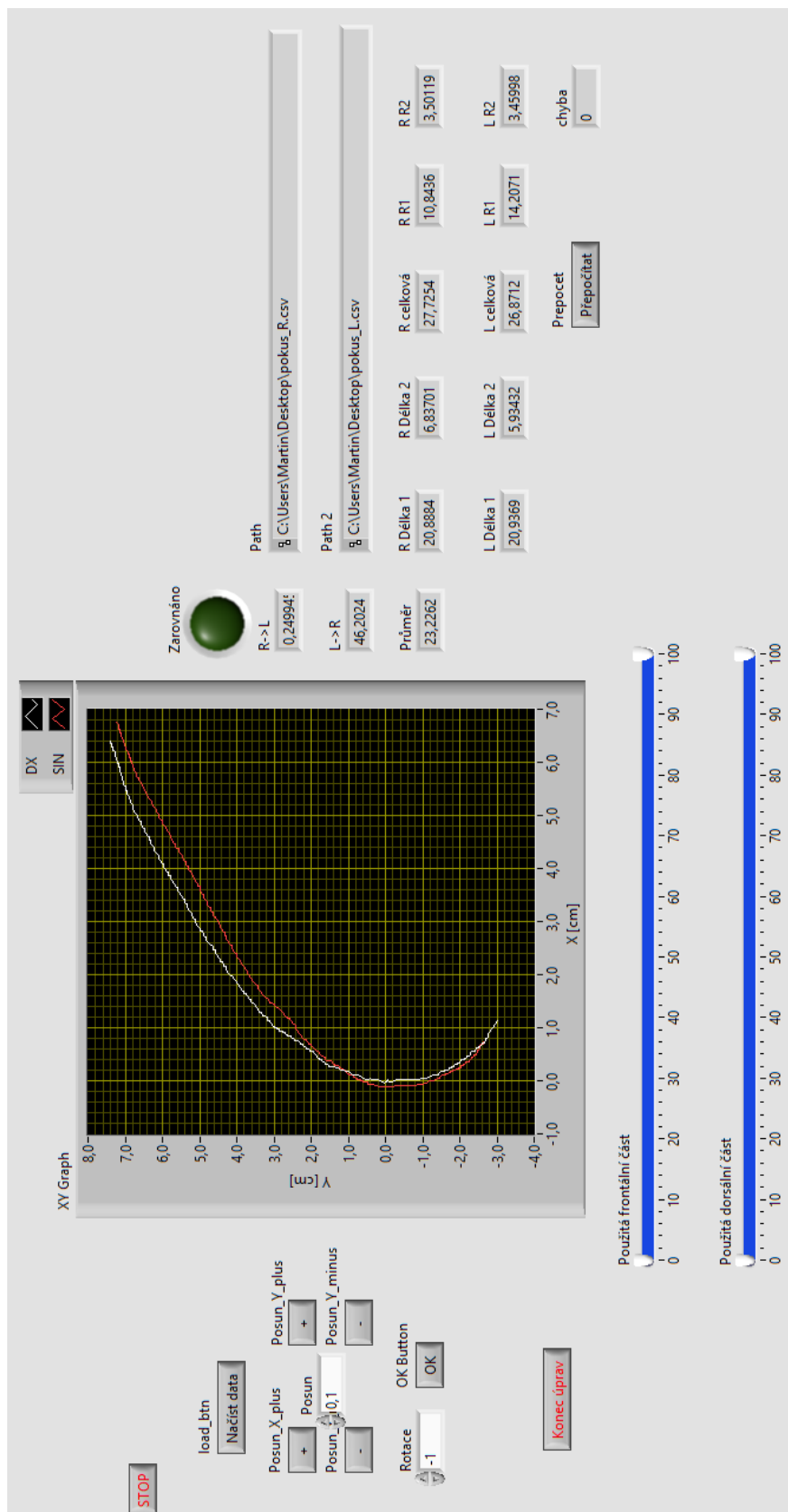
Jak je vidět na obrázku 4.11, tak uživatelské rozhraní je rozděleno do čtyř sekcí. První (vlevo) obsahuje ovládací prvky pro pohyb levé křivky a prvky pro kontrolu běhu programu (posun v sekvenci). Přičemž posuny křivky mají společné číselné pole, což má tu výhodu, že pokud uživatel provede chybný krok, tak se může stiskem jednoho tlačítka vrátit zpět, aniž by musel přepisovat hodnoty v číselných polích.

V centrální části nahoře je umístěn graf. Ten je doplněn o legendu, kde DX odpovídá pravé straně a SIN levé. Graf navíc umožňuje po stisku pravého tlačítka export do zjednodušeného obrázku pro případný tisk. Dole pod grafem se pak nachází posuvníky pro nastavení rozsahu prokládání kružnice. Uživatel si však musí dávat pozor na dvě věci. První z nich je, že interval mezi posuvníky musí mít modrou barvu. Pokud je šedý a naopak okolí je vybarveno, tak má posuvníky přehozeny a program patrně nedává správné hodnoty. Druhou věcí je uvědomit si, že posuvníky pracují s indexy, tedy křivka začíná na horním posuvníku vždy vlevo a postupuje doprava kde má křivka bod nejbližší k nule. Pro zadní část platí, že bod nejbližší k nule začíná vlevo a konec křivky je vpravo.

Vpravo od grafu se nachází napočítané hodnoty a signalizace zarovnání s vypsanými cestami k souborům s daty. Data jsou v programu zobrazena v podobném pořadí jako jsou ukládána v souboru s výsledky. Rovněž je zde umístěno tlačítko pro přepočet poloměrů.

## 4.8 Výstupní data ke statistickému zhodnocení

Výstupní data obsahují údaje uvedené v sekci Proložení kružnice a uložení výsledků. Tedy název vstupního datového souboru, vzdálenosti křivek, délky křivek včetně délek po sekcích a poloměry na obou křivkách. Data jsou formátována podobným způsobem, jako u dat vstupních, avšak tyto data jsou doplněny o hlavičku. To spolu s dodržáním formátu CSV umožňuje požit běžný tabulkový procesor (například LibreOffice Calc, Microsoft Office Excel). Zároveň je tak zaznamenán i význam jednotlivých polí, uživatel tak nemusí hledat v dokumentaci.



Obrázek 4.11: Uživatelské rozhraní

## 5. Závěr

Během této bakalářské práce byly ve shodě se zadáním realizovány programy na rozpoznání požadované křivek pánve ze snímků CT a na jejich vyhodnocení. Rovněž byly výstupy těchto programů průběžně kontrolovány, zda podávají relevantní a správné informace. Přičemž u finální verze programů nebyly zjištěny nesrovnalosti mezi vstupními a výstupními daty které by naznačovaly přítomnost chyby ve výpočetních algoritmech. Ve shodě se zadáním také byla vytvořena dokumentace k vytvořenému softwaru, kterou lze nalézt v přílohách k této práci.

Ačkoliv bylo dosaženo všech cílů vyplývajících ze zadání, tak lze software dále rozvíjet. Kupříkladu určitý potenciál k dalšímu rozvoji spatřuji v automatické identifikaci objektů v snímku CT. Kde je nyní potřeba, kvůli velké variabilitě lidského těla, zásah obsluhy. Je však nutné připomenout, že se jedná o natolik náročnou úlohu z oblasti počítačového vidění, že by její realizace obsáhla samostatnou práci. Z dalších možností k dalšímu rozvoji lze jmenovat například automatické vyhodnocování měřítka snímků nebo automatické sesazování křivek z levé a pravé strany. Poslední jmenovaná možnost by však byla svou povahou taktéž náročnější.

Vytvořené programy jsou v době psaní této práce využívány pro získání dat ke statistickému vyhodnocení prostorových dispozic pánve v rámci diplomového projektu: „HUNĚK, Martin. *Statistical data processing of inner pelvis anatomy structure from CT images*. Liberec, 2014. Diploma project. Technical university of Liberec, Faculty of Mechatronics, Informatics and Interdisciplinary Studies. Supervisor Ing. Martin Kysela. “, který tak na tuto práci volně navazuje.

## Literatura

- [1] ASCII. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001–2014 [cit. 2014-02-22]. Dostupné z: <http://cs.wikipedia.org/wiki/ASCII>
- [2] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, 2008, xvii, 555 s. ISBN 978-0-596-51613-0.
- [3] DUINTJER TEBBENS, Jurjen, Iveta HNĚTYNKOVÁ, Martin PLEŠINGER, Zdeněk STRAKOŠ a Petr TICHÝ. *Analýza metod pro maticové výpočty: základní metody*. Vyd. 1. Praha: Matfyzpress, 2012, xvi, 308 s. ISBN 978-80-7378-201-6.
- [4] HLAVÁČ, Václav. *Zpracování signálů a obrazů*. 1. vyd. Praha: Vydavatelství ČVUT, 2002, 220 s. ISBN 80-010-2114-9.
- [5] HOMOLKA, František. ČVUT FBMI. *Použití technických norem ve zdravotnictví - oblast zkušebnictví a působnosti EZÚ Praha*. Kladno, 2007. Dostupné z: <http://www.fbmi.cvut.cz/esf-realizovane/www.fbmi.cvut.cz/e/pouziti-technicky-norem-ve-zdravotnictvi-oblast-zkusebnictvi-a-pusobnosti-ezu-praha/1847.pdf>
- [6] CHALOUPKA, Josef. Filtrace šumu, hledání hran. In: *Přednášky z předmětu: Snímání, digitalizace a zpracování obrazových dat* [online]. Liberec: Technická univerzita v Liberci, 2013 [cit. 2013-11-20]. Dostupné z: <https://www.ite.tul.cz/vyuka/mod/resource/view.php?id=831>
- [7] CHALOUPKA, Josef. Detekce hran, rekonstrukce obrazu. In: *Přednášky z předmětu: Snímání, digitalizace a zpracování obrazových dat* [online]. Liberec: Technická univerzita v Liberci, 2013 [cit. 2013-11-20]. Dostupné z: <https://www.ite.tul.cz/vyuka/mod/resource/view.php?id=831>

- [8] JAKSCH, Ivan. Základy digitálního obrazu: Vektorová a rastrová grafika. In: *Přednášky z předmětu Zobrazovací systémy* [online]. Liberec: Technická univerzita v Liberci, 2012 [cit. 2013-11-20]. Dostupné z: [http://www.fm.tul.cz/esf0247/download.file.php/ESF\\_zakl.digit.obr.pdf?id=838](http://www.fm.tul.cz/esf0247/download.file.php/ESF_zakl.digit.obr.pdf?id=838)
- [9] Least Squares Circle Calculator. HAD2KNOW. *Had2Know.com* [online]. 2010–2014 [cit. 2014-02-22]. Dostupné z: <http://www.had2know.com/academics/best-fit-circle-least-squares.html>
- [10] MIČKA, Pavel. *ASCII tabulka*. [online]. 2013 [cit. 2014-02-22]. Dostupné z: <http://www.algoritmy.net/article/pdf/89/asciiTable>
- [11] NATIONAL INSTRUMENTS. *LabVIEW Fundamentals* [online]. Austin: National Instruments, 2005, 165 s., August 2005 [cit. 2014-03-23]. 374029A-01. Dostupné z: <http://www.ni.com/pdf/manuals/374029a.pdf>
- [12] NATIONAL INSTRUMENTS. *NI Vision Concepts Manual* [online]. Austin: National Instruments, 2008, 414 s., June 2008 [cit. 2014-03-23]. 372916G-01. Dostupné z: <http://www.ni.com/pdf/manuals/372916g.pdf>
- [13] NATIONAL INSTRUMENTS. *LabView: Getting Started with LabVIEW* [online]. Austin: National Instruments, 2012, 2012-06 [cit. 2013-11-20]. Dostupné z: <http://www.ni.com/pdf/manuals/373427h.pdf>
- [14] PLÍVA, Zdeněk a Jindra DRÁBKOVÁ. *Metodika zpracování diplomových, bakalářských a vědeckých prací na FM TUL*. Liberec: TU v Liberci, 2007. ISBN:978-80-7372-189-3.
- [15] PLÍVA, Z., J. DRÁBKOVÁ, J. KOPRNICKÝ a L. PETRŽÍLKA. *Metodika zpracování bakalářských a diplomových prací*. 2. upravené vydání. Liberec: Technická univerzita v Liberci, FM, 2014. ISBN 978-80-7494-049-1.
- [16] RUSS, John C, J. RUSS. *Introduction to image processing and analysis*. Boca Raton: CRC Press, 2008, 355 s. ISBN 08-493-7073-6.
- [17] SATRAPA, Pavel. *LaTeX pro pragmatiky*. Liberec: Technická univerzita v Liberci, 2011. Dostupné z: <http://www.nti.tul.cz/satrapa/docs/latex/>

- [18] SEUL, Michael. *Practical algorithms for image analysis: description, examples and code*. Vyd. 1. New York: Cambridge University Press, 2005, 295 s. ISBN 05-216-6065-3.
- [19] Šonka M., V. Hlaváč, R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 1998, ISBN 0-534-953-93.
- [20] ŠRÁM, J., S. TALLER, R. LUKÁŠ a L. ENDRYCH. Užití Omega dlahy při stabilizaci zlomenin acetabula - první zkušenosti. In: *ACTA CHIRURGIAE ORTHOPAEDICAE ET TRAUMATOLOGIAE ČECHOSL*. 2013, č. 80, 118 - 124. Dostupné z: <http://www.achot.cz/detail.php?stat=617>
- [21] TALLER, S., J. ŠRÁM, R. LUKÁŠ, M. KŘIVOHLÁVEK: Zlomeniny pánevního kruhu a acetabula operované přístupem dle Stoppy. In: *Acta Chir. orthop. Traum. Čech.*, 77: 93-98, 2010.
- [22] Vyhledávání v zákonech - portál veřejné správy. MINISTERSTVO VNITRA ČR. *Portál veřejné správy* [online]. 2012 [cit. 2014-01-02]. Dostupné z: <http://portal.gov.cz/app/zakony/?path=/portal/obcan/>
- [23] WILKINS, David. R. *Getting Started with LaTeX*. SCHOOL OF MATHEMATICS TRINITY COLLEGE, Dublin. [online]. 2. vydání. Dublin: School of Mathematics Trinity College, 1995 [cit. 2014-02-22]. Dostupné z: <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>
- [24] WOHLER, Christian. *3d computer vision: efficient methods and applications*. 2nd ed. London: Springer, 2013, 381 s. ISBN 978-1-4471-4149-5.
- [25] Zdravotnické prostředky, Státní ústav pro kontrolu léčiv. STÁTNÍ ÚSTAV PRO KONTROLU LÉČIV. *SÚKL - Ostatní činnosti* [online]. 2010 [cit. 2014-01-02]. Dostupné z: <http://www.sukl.cz/zdravotnicke-prostredky>



## A. Obsah CD

CD/	
bin.....	Spustitelné soubory
_ extrakce	
_ extrakce.exe .....	Program pro extrakci křivky
_ ...	
_ srovnani	
_ srovnani.exe .....	Program pro srovnání získaných křivek
_ ...	
img.....	Obrázky použité v práci
_ ...	
pdf .....	Dokumenty ve formátu PDF
_ BP.pdf.....	Text této práce
_ dok1.pdf.....	Dokumentace k programu pro extrakci křivky
_ dok2.pdf.....	Dokumentace k programu pro srovnání získaných křivek
_ ...	
src .....	Zdrojové kódy k programům
_ Circular fit.vi.....	Funkce pro prokládání kružnice
_ extrakce.vi .....	Program pro extrakci křivky
_ srovnani.vi.....	Program pro srovnání získaných křivek
tex.....	Zdrojové kódy k dokumentům v $\text{\LaTeX}$ u
_ BP.tex.....	Zdrojový kód k této práci
_ dok1.tex .....	Zdrojový kód k dokumentaci extrakce křivky
_ dok2.tex.....	Zdrojový kód k dokumentaci srovnání získaných křivek
_ ...	
obsah.pdf .....	Tato příloha
obsah.pdf.sig.....	Podpis obsahu

Uvedené soubory jsou podepsány PGP klíčem:

Martin Huněk <Martin.Hunek@tul.cz>

Martin Huněk <hunekm@gmail.com>

(2048 bits RSA; Key-ID: 511DE55C; Created: 2012-04-21)

SHA-1: 0D31 4F18 9816 CA4F C143 4FE0 B510 65EA 511D E55C

## **B. Dokumentace k programu pro extrakci křivky**



TECHNICKÁ UNIVERZITA V LIBERCI  
Ústav zdravotnických studií

*Dokumentace k programu  
pro extrakci křivky*

**Bc. Martin Huněk**

6. 4. 2014

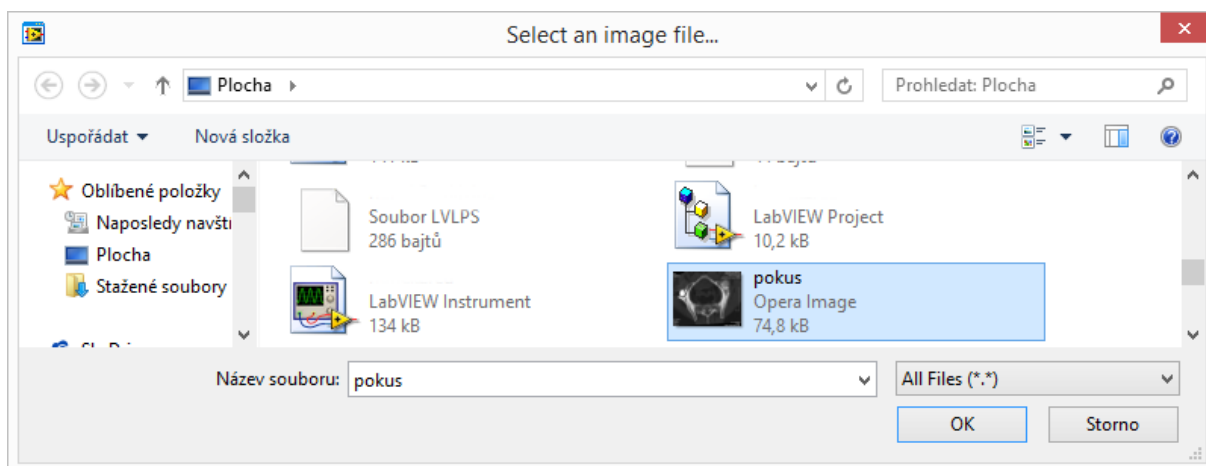


## 1 Systémové požadavky

**Procesor:** x86 s podporou SSE2 (Intel Pentium 4 / AMD Athlon 64 a novější)  
**RAM:** minimálně 256MiB  
**Rozlišení:** minimálně 1024x786  
**OS:** Windows 8.1/8/7/Vista 32b nebo 64b, XP SP3 32b  
**SW:** NI LabVIEW Run-Time Engine 2012

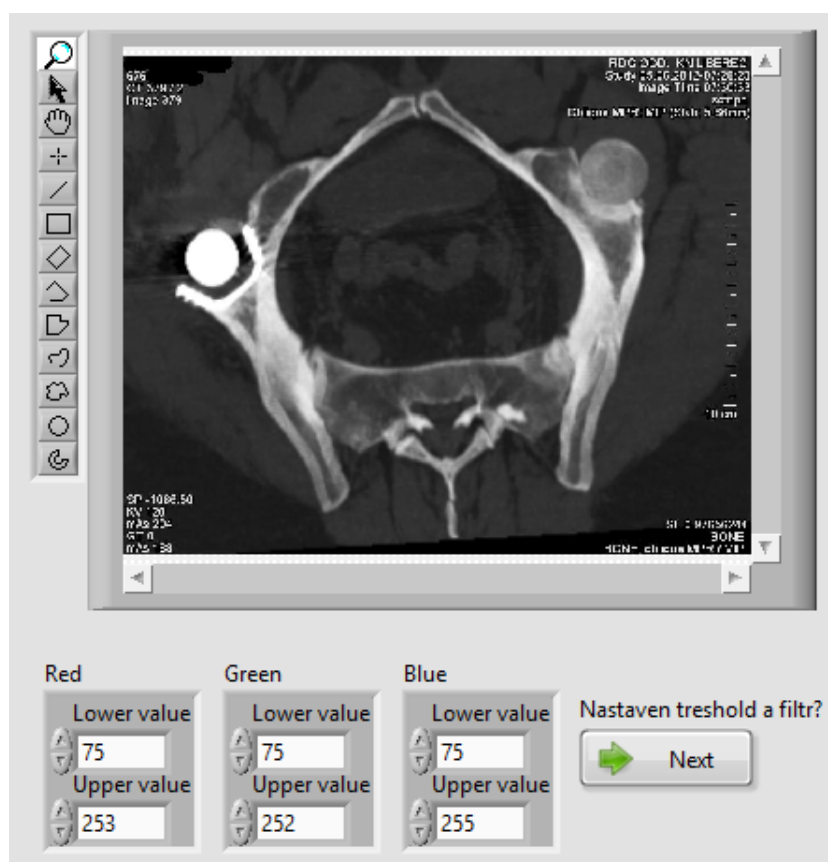
## 2 Návod k použití

Po spuštění programu „extrakce.exe“ je nejprve zobrazen souborový dialog jak ukazuje obrázek 1. V něm vyberte snímek pánve ve formátu JPEG (přípona .jpg nebo .jpeg). Program je uzpůsoben ke zpracování snímků pánve v takzvané počítačovým tomografem definované (CTD) projekci pánevního vchodu. Ačkoliv může fungovat i na jiných snímcích, tak jeho funkčnost na nich není plně zaručena.



Obrázek 1: Dialog pro otevření souborů

Po načtení souboru se objeví v okně náhledu původní snímek a v hlavním pracovním okně se objeví snímek prahovaný podle aktuálního nastavení (obrázek 3). Doporučené nastavení se pohybuje okolo přednastavené hodnoty 75 pro každou z barev (obrázek 2). Pokud je některá okolní tkáň vyhodnocena jako kost, zvýšte spodní hodnotu prahového



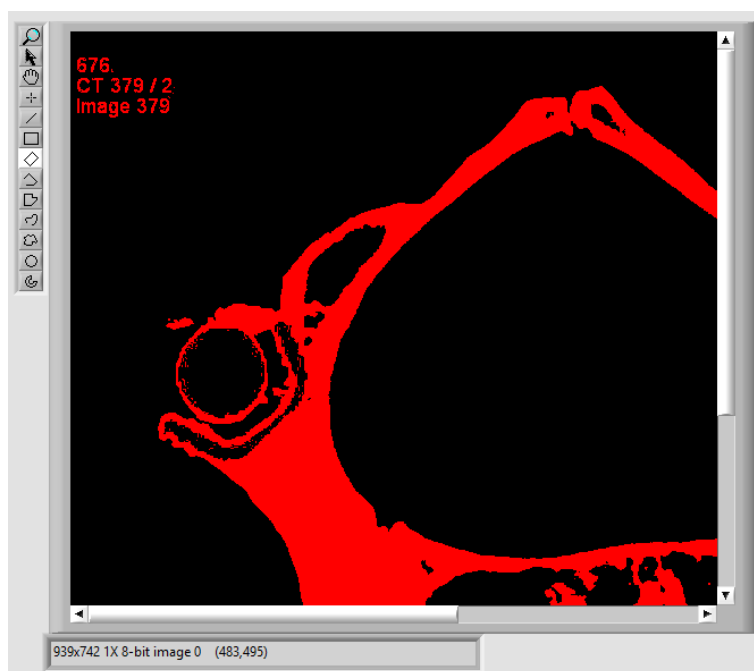
Obrázek 2: Nastavení prahového filtru

filtru. Pokud naopak část kosti v upraveném snímku chybí, tak tuto hodnotu snižte a nastavte horní hodnoty na 255.

Jakmile je filtr nastaven, je po stisku tlačítka „Next“ možné přistoupit k výběru křivky a středu acetabula (bod na křivce nejblíže středu acetabula). Je doporučeno nejdříve najít střed acetabula a až poté zvolit oblast s křivkou i když program nevyžaduje dodržení pořadí těchto kroků.

Z důvodu co největší jednoznačnosti je doporučeno k hledání středu acetabula použít úsečku, jak to ukazuje obrázek 4. Tu vytvoříte pomocí pátého nástroje shora v nástrojové liště a následným tažením od středu hlavice femuru na zkoumané straně do protějšího SI skloubení. Tam, kde úsečka opustí kost směrem od femuru se nachází hledaný bod.

Výběr bodu je proveden dvojitým kliknutím a je signalizován pomocí změny zobrazených souřadnic nad tlačítkem „OK“. Tento výběr bodu doporučuji dělat pomocí nástroje kurzoru jak ukazuje obrázek 5. Nestane se tak, že v případě, že se vám nepovede

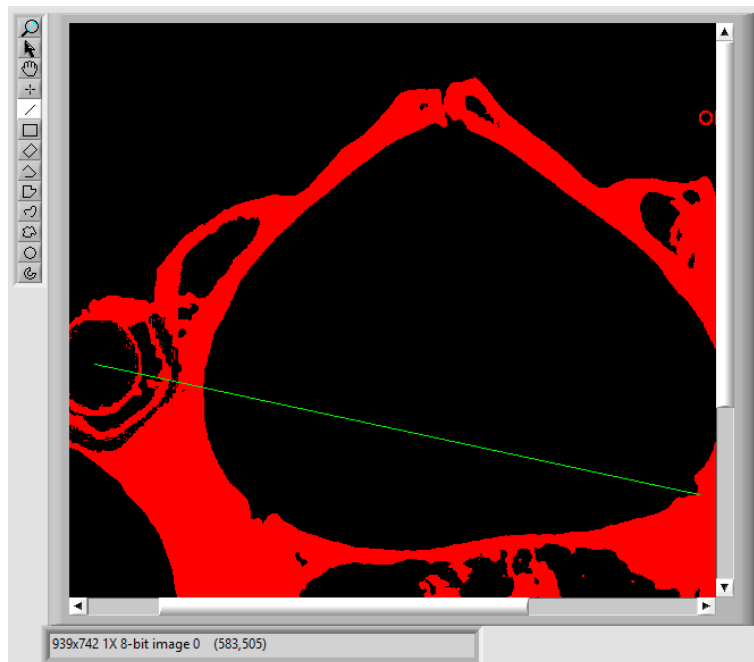


Obrázek 3: Snímek po prahování

bod označit, tak nepřijdete o vytvořenou úsečku.

Následně můžete přistoupit k výběru oblasti se křivkou. Její výběr lze udělat pouze pomocí nástroje otočného obdélníku a jeho následným tažením. Na jiný obrazec program nereaguje. Postup ukazuje obrázek 6, kde si všimněte hlavně zvoleného nástroje.

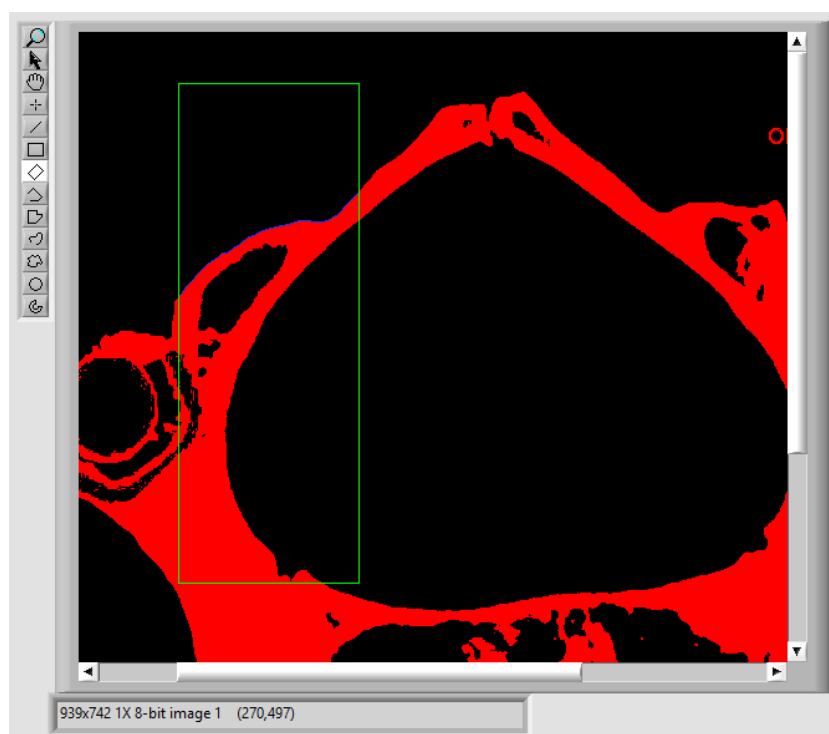
Otočení a změna velikosti se provádí opět tažením, tentokrát však v blízkosti hran oblasti pro velikost a středů hran pro otočení. Výsledek by měl vypadat jako ten na obrázcích 7 a 8. Přičemž výsledná křivka je znázorněna modrou barvou jak v náhledu, tak v pracovním okně.



Obrázek 4: Vytvoření pomocné úsečky



Obrázek 5: Nástrojová lišta přepnutá na ukazatel



Obrázek 6: Vytvoření otočného obdélníku





### 3 Řešení problémů

1. Program neoznačuje správnou křivku.
  - Zúžete oblast tak, aby byla co nejmenší a zároveň pokrývala celou křivku.
  - Pokud program označuje příliš krátké křivky zvětšete parametr „Min. Length“.
  - Změňte parametr „Contour Selection“ na jednu z prvních dvou možností.
2. Program neumožňuje cokoli dělat v hlavním okně. Cokoli uělám okamžitě zmizí.
  - Nepotvrdili jste nastavení prahového filtru. Zkontrolujte ho a stiskněte „Next“.
3. Mám označenou křivku, ale nejde stisknout tlačítko „OK“.
  - Nebyl zvolen bod středu acetabula. Zvolte jej.
4. Program vůbec nereaguje.
  - Program musí být spuštěn. V horní liště se nachází symbol pravé šipky. Pokud má bílou barvu, program se nevykonává. Stiskněte ji.

### 4 Zdrojový kód

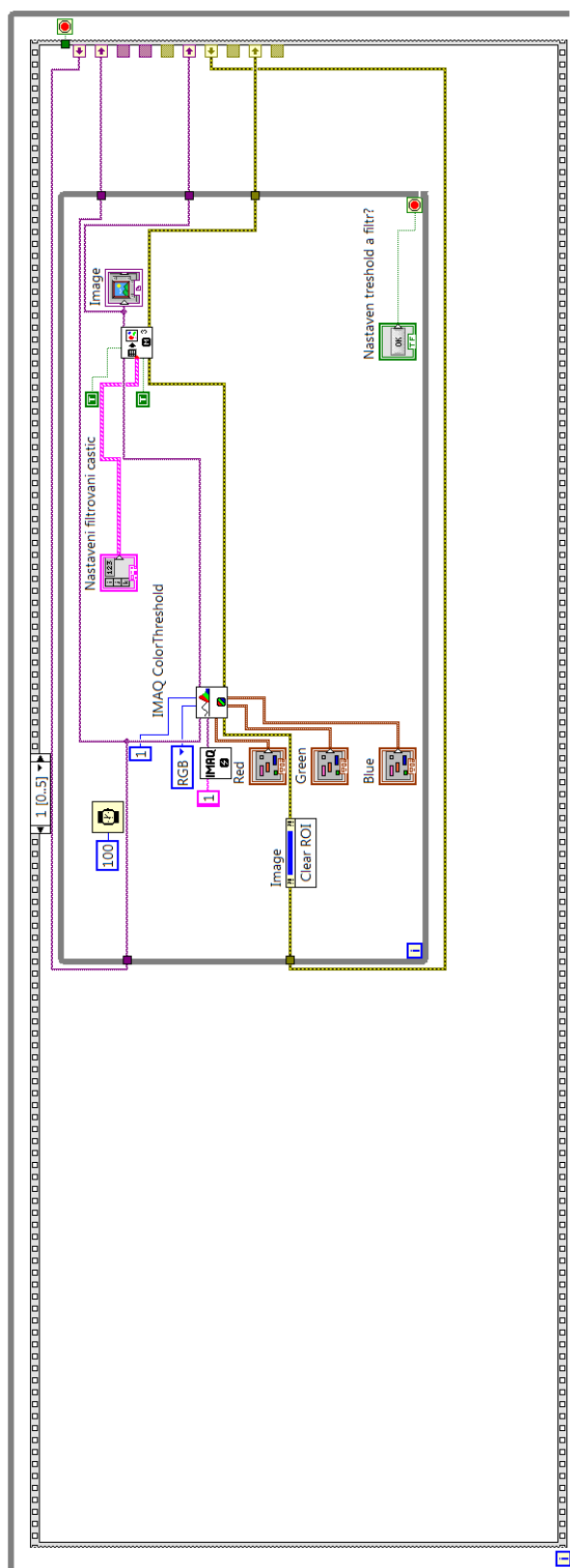
Kompletní zdrojový kód v prostředí je k vidění na obrázcích 9 až 16.

### 5 Závěrečná ustanovení

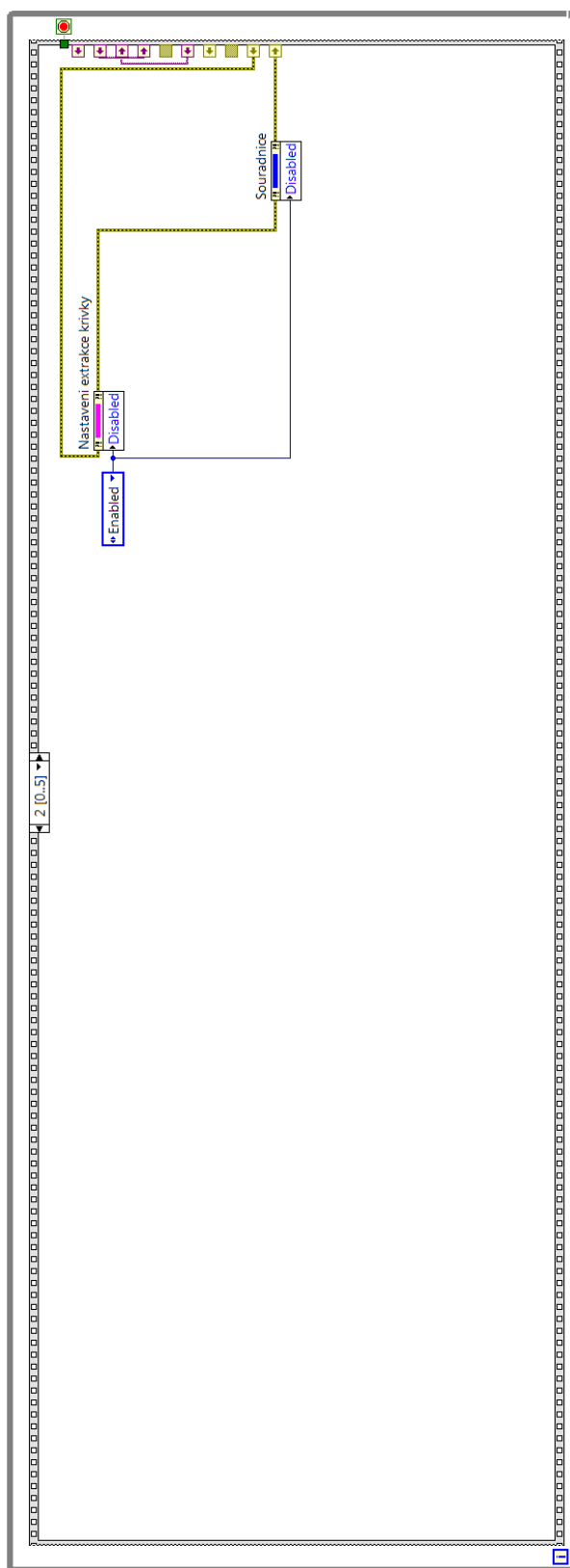
Tento dokument je přílohou k bakalářské práci: HUNĚK, Martin. *Realizace programu pro optimalizace tvaru a velikosti vnitřních pánevních dlah z řezů CT*. Liberec, 2014. Bakalářská práce. Technická univerzita v Liberci, Ústav zdravotnických studií. Vedoucí práce MUDr. Jaroslav Šrám.

Šíření tohoto dokumentu je zakázáno včetně vytváření odvozených děl z přiložených zdrojových kódů výjma případů kladného písemného stanoviska autora nebo Technické univerzity v Liberci (v mezích stanovených zákonem 121/2000Sb. O právu autorském, ve znění pozdějších předpisů).

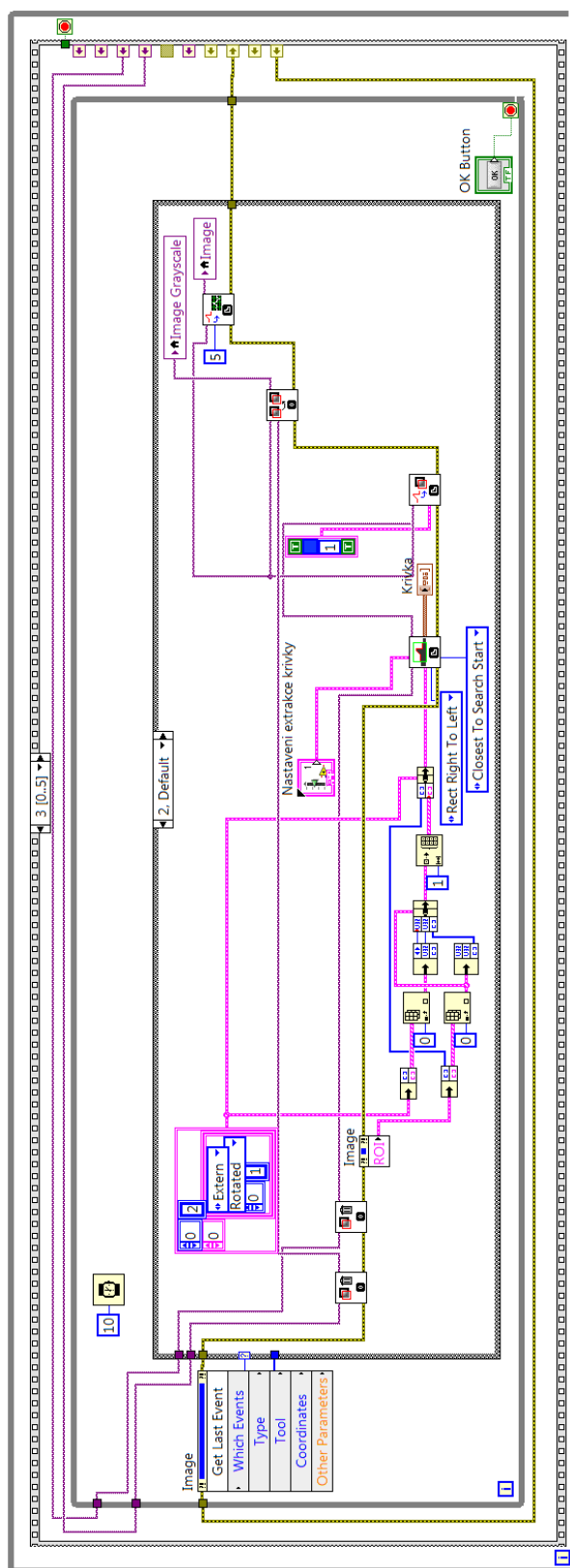




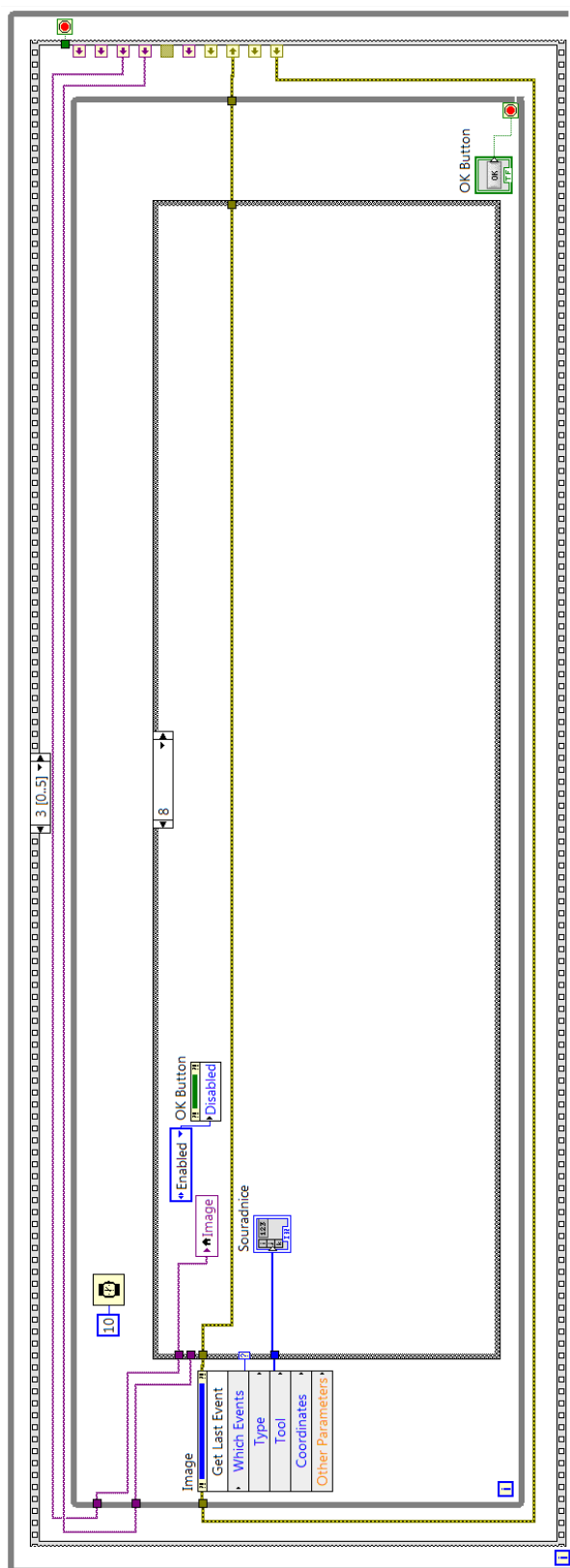
Obrázek 10: Prahování obrazu a filtrování částic



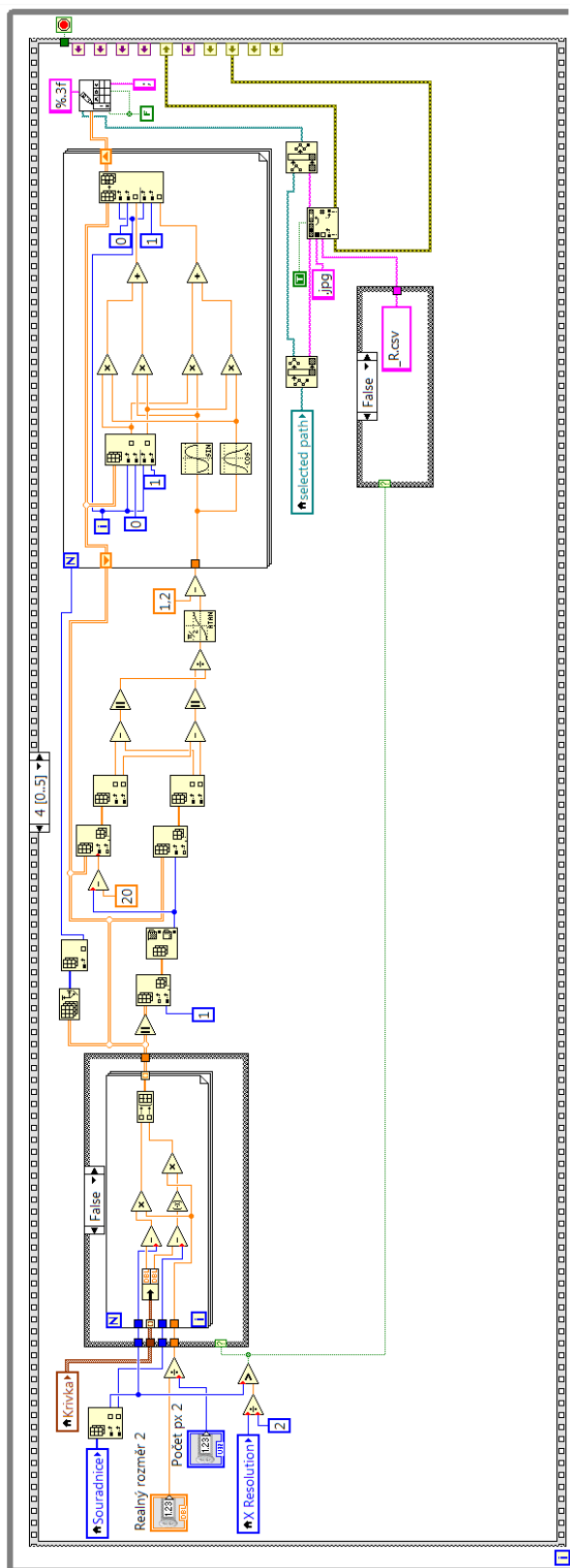
Obrázek 11: Odblokování pracovního okna



Obrázek 12: Extrakce křivky

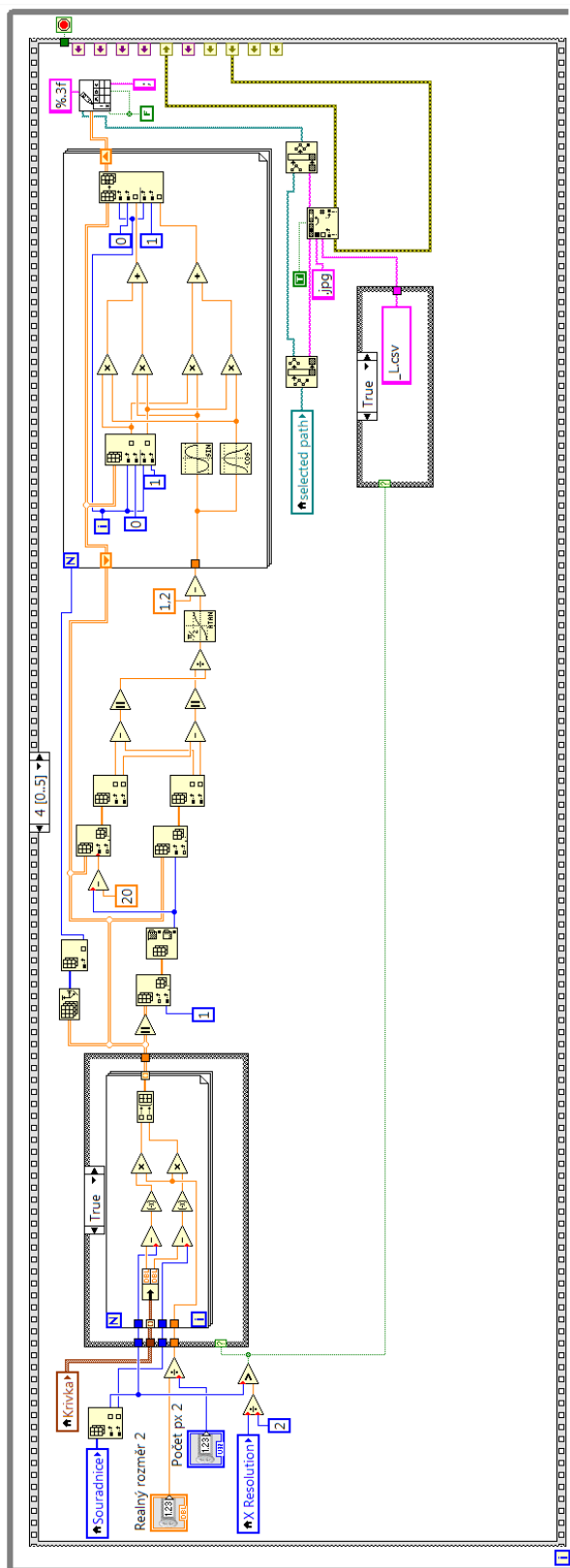


Obrázek 13: Uložení souřadnic středu acetabula



Obrázek 14: Otočení a uložení pravé křivky





Obrázek 15: Otočení a uložení levé křivky



## C. Dokumentace k programu pro srovnání získaných křivek



TECHNICKÁ UNIVERZITA V LIBERCI  
Ústav zdravotnických studií

*Dokumentace k programu  
pro srovnání získaných křivek*

**Bc. Martin Huněk**

6. 4. 2014

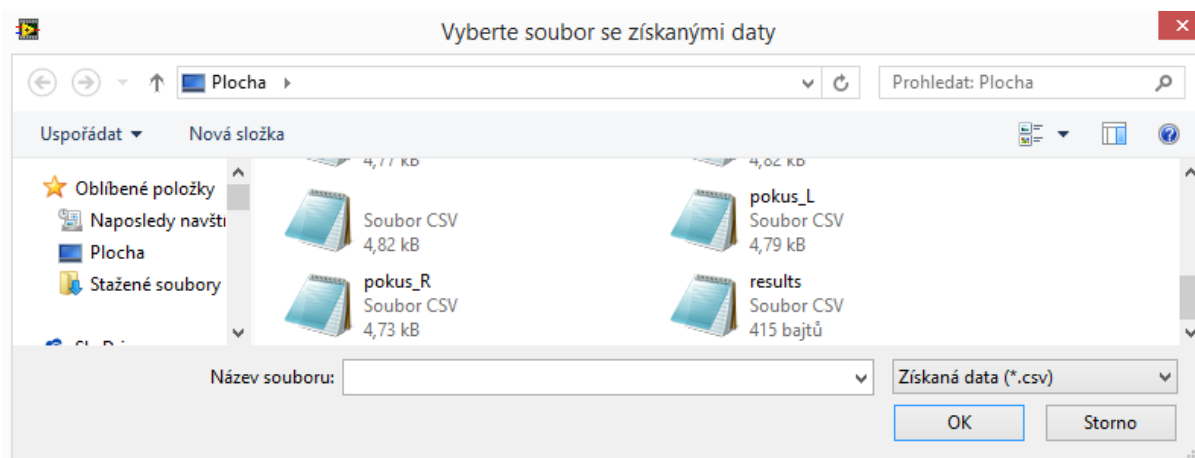


## 1 Systémové požadavky

**Procesor:** x86 s podporou SSE2 (Intel Pentium 4 / AMD Athlon 64 a novější)  
**RAM:** minimálně 256MiB  
**Rozlišení:** minimálně 1024x786  
**OS:** Windows 8.1/8/7/Vista 32b nebo 64b, XP SP3 32b  
**SW:** NI LabVIEW Run-Time Engine 2012

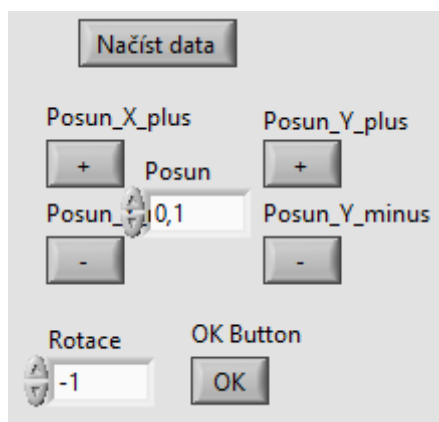
## 2 Návod k použití

Po spuštění programu „srovnani.exe“ je po stisku tlačítka „Načíst data“ zobrazen souborový dialog jak ukazuje obrázek 1. V něm vyberte data pánve ve formátu CSV (přípona .csv). Data se musí skládat ze dvou souborů z nichž název jednoho z nich musí končit na „R.csv“ a druhého na „L.csv“.



Obrázek 1: Dialog pro otevření souborů

Tím dojde k zobrazení křivek v bodovém grafu (ukazuje obrázek 3) a zároveň je umožněno hýbat s levou křivkou oproti pravé. To lze provést zapsáním hodnoty o kterou chcete křivku posunout nebo otočit (pohyb se zadává ve stejných jednotkách jako je měřítko, rotace se zadává ve stupních). Následně je pak nutné stisknout tlačítko pro pohyb patřičným směrem nebo pro rotaci (jak je vidět na obrázku 2). Opačný směr rotace lze docílit záporným znaménkem.



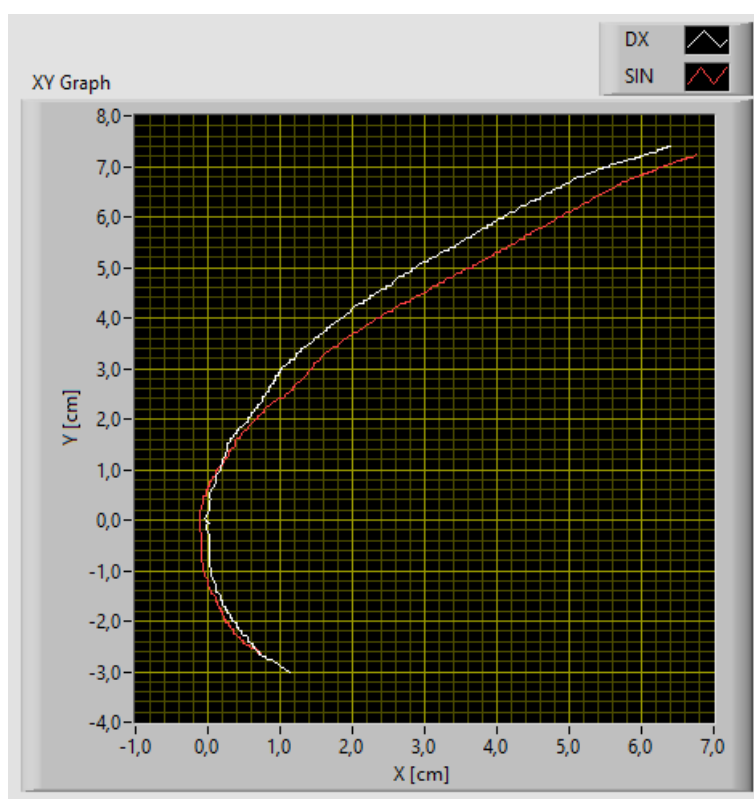
Obrázek 2: Ovládací prvky pro pohyb křivkou

Obrázek 4 ukazuje vypočtené hodnoty a signalizaci zarovnání. Signalizaci zarovnání berte pouze orientačně. Vždy jde o to zarovnat na sebe obě křivky co nejpřesněji, ale v některých případech se může stát, že se vám nepodaří na sebe křivky dokonale zarovnat a signalizaci zarovnání tak rozsvítit. Všechny uvedené hodnoty se přepočítávají automaticky při pohybu křivky až na poloměry. Ty jsou pro svou větší výpočetní náročnost počítány pouze na vyžádání uživatele nebo před uložením. Ruční vyvolání výpočtu poloměrů lze docílit tlačítkem „Přepočítat“.

V případě, že je získaná křivka zahnutá pouze na určitém úseku v přední nebo zadní části, je možné tento úsek ručně vybrat. Na obrázku 5 jsou vidět posuvníky kterými je výběr prováděn. Modrý interval mezi posuvníky značí interval ve kterém je vypočítáván poloměr, šedá část je při výpočtu ignorována. Při označování intervalu je potřeba dávat pozor na skutečnost, že nula na posuvníku vždy odpovídá bodům blíže k začátku souboru (u křivek získaných programem „extrakce.exe“ je to horní konec křivky).


V momentě, kdy jste s výsledky spokojeni ukončete úpravy tlačítkem „Konec úprav“, jak je zobrazeno na obrázku 6. Tím dojde k vyvolání dialogu z obrázku 7, kdy volbou „Ne“ dojde k zahazení dat a opětovnému startu programu a datový soubor tak není zaplavován možnými nezdařilými pokusy. Naopak volba „Ano“ vyvolá dialog 8 kde můžete vybrat soubor, kam se data uloží. V případě, že soubor neexistuje dojde k jeho vytvoření, pokud existuje, tak se záznam přidá na konec souboru.

Implicitní název souboru je „results.csv“ a implicitní cesta k souboru vede do stejného adresáře odkud byla získána data. Jako uživatel můžete zvolit jiný název a umístění, avšak to nedoporučuji neboť by vás tato změna umístění čekala při každém ukládání dat.



Obrázek 3: Zobrazení obou křivek

Zarovnáno



R->L  
0,24994%

Path  
C:\Users\Martin\Desktop\pokus\_R.csv

L->R  
46,2024

Path 2  
C:\Users\Martin\Desktop\pokus\_L.csv

Průměr	R Délka 1	R Délka 2	R celková	R R1	R R2
23,2262	20,8884	6,83701	27,7254	10,8436	3,50119
	L Délka 1	L Délka 2	L celková	L R1	L R2
	20,9369	5,93432	26,8712	14,2071	3,45998

Prepočet  
Přepočítat

chyba  
0

Obrázek 4: Vypočtené hodnoty a signalizace zarovnání

Použitá frontální část

0 10 20 30 40 50 60 70 80 90 100

Použitá dorsální část

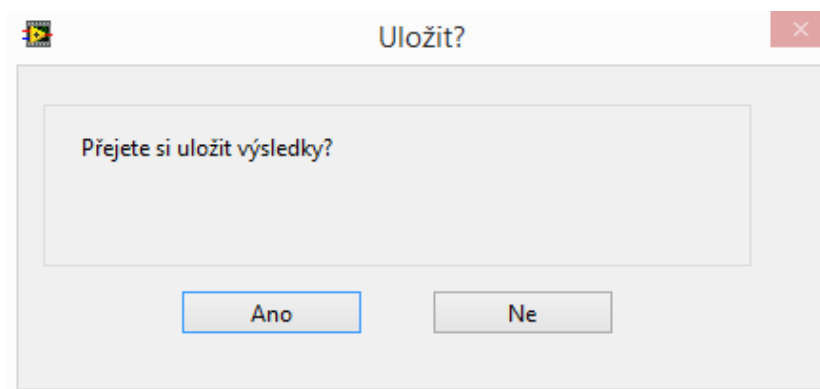
0 10 20 30 40 50 60 70 80 90 100

Obrázek 5: Posuvníky pro specifikování rozsahu pro výpočet poloměru

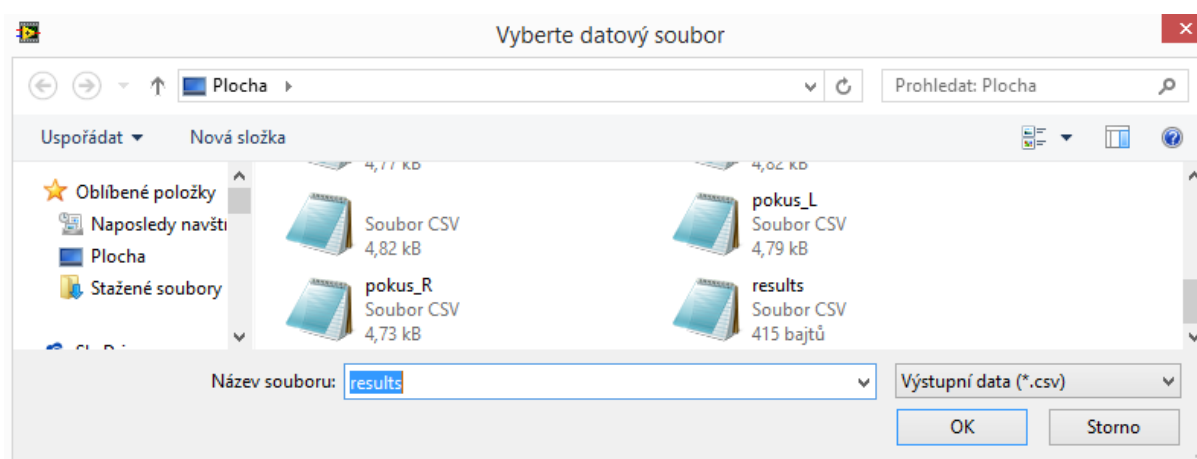
Konec úprav

Obrázek 6: Tlačítko pro konec úprav





Obrázek 7: Dialog uložení dat



Obrázek 8: Výběr souboru pro uložení dat

### 3 Řešení problémů

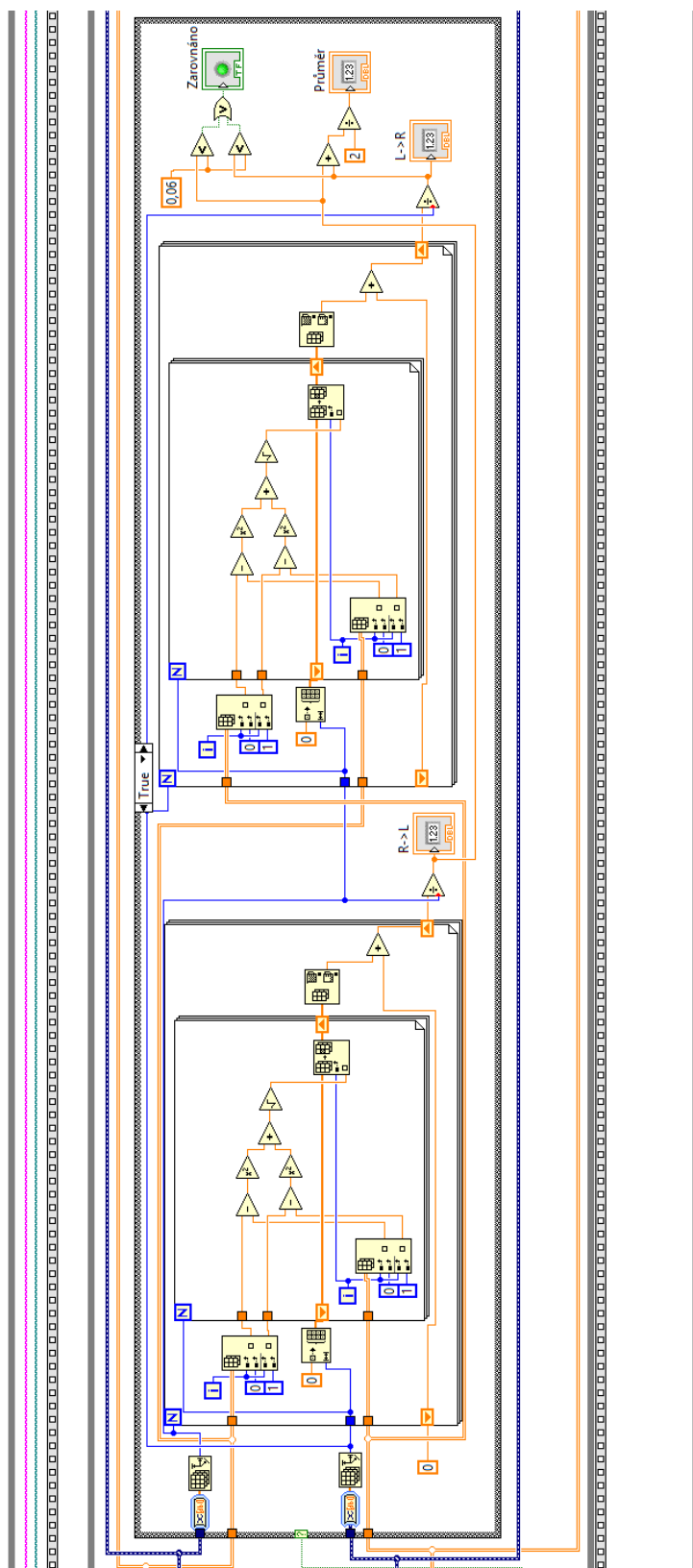
1. Program nenačítá soubor s daty.
  - Zkontrolujte zda je soubor s daty správně pojmenován.
  - Ve stejném adresáři musí být oba soubory (jak končící \_R.csv, tak \_L.csv).
  - V každém souboru by měl být alespoň jeden řádek o dvou číslech oddělených středníkem.
2. Po stisku tlačítek pro pohyb se nic neděje.
  - V příslušném poli pro parametry pohybu nesmí být nuly.
3. Křivky nelze zarovnat tak, aby svítila kontrolka „Zarovnáno“.
  - Ne všechny křivky jde takto zarovnat. Pokud jsou křivky příliš rozdílné, tak se pokuste křivky zarovnat aby jeden z parametrů („R->L“ nebo „L->R“) byl co nejmenší
4. Program vůbec nereaguje.
  - Program musí být spuštěn. V horní liště se nachází symbol pravé šipky. Pokud má bílou barvu, program se nevykonává. Stiskněte ji.
5. Program nepočítá poloměry.
  - Poloměry se nepočítají automaticky, stiskněte tlačítko „Přepočítat“.
  - Modrá část intervalu musí být mezi posuvníky, pokud není přesuňte jeden přes druhý.

### 4 Zdrojový kód

Kompletní zdrojový kód v prostředí je k vidění na obrázcích 9 až 16.



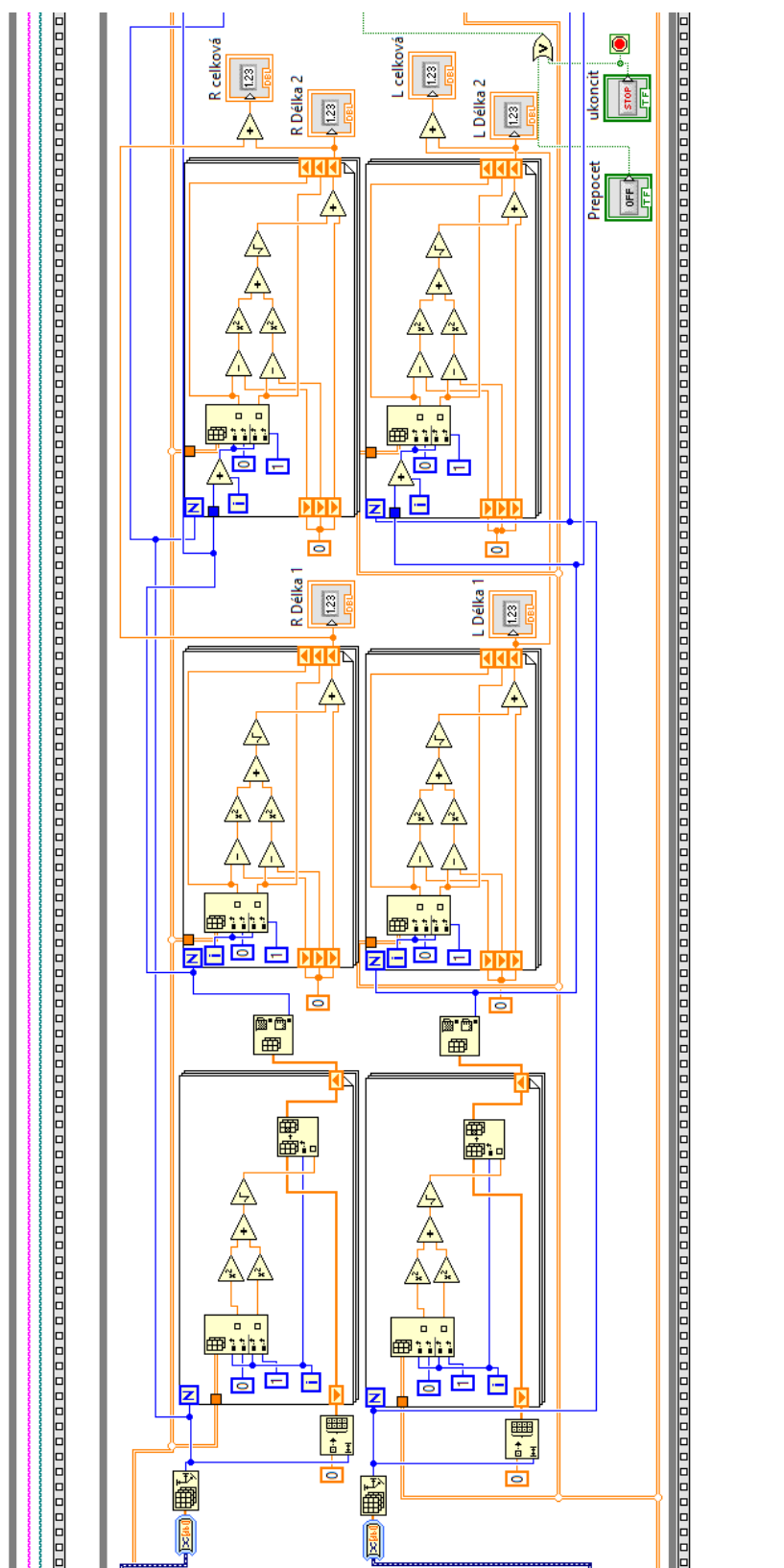




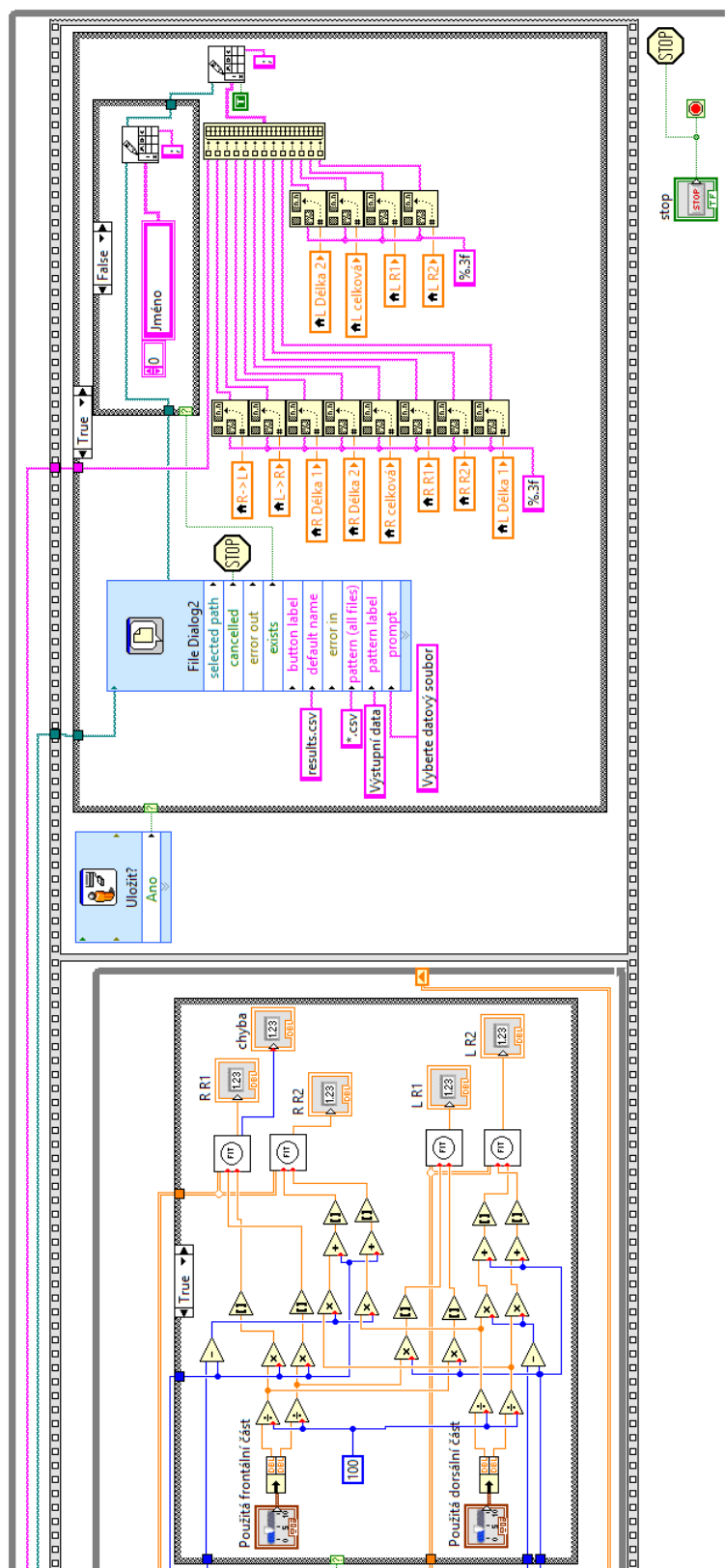
Obrázek 11: Načtení obrazu a zobrazení náhledu



Obrázek 12: Načtení obrazu a zobrazení náhledu



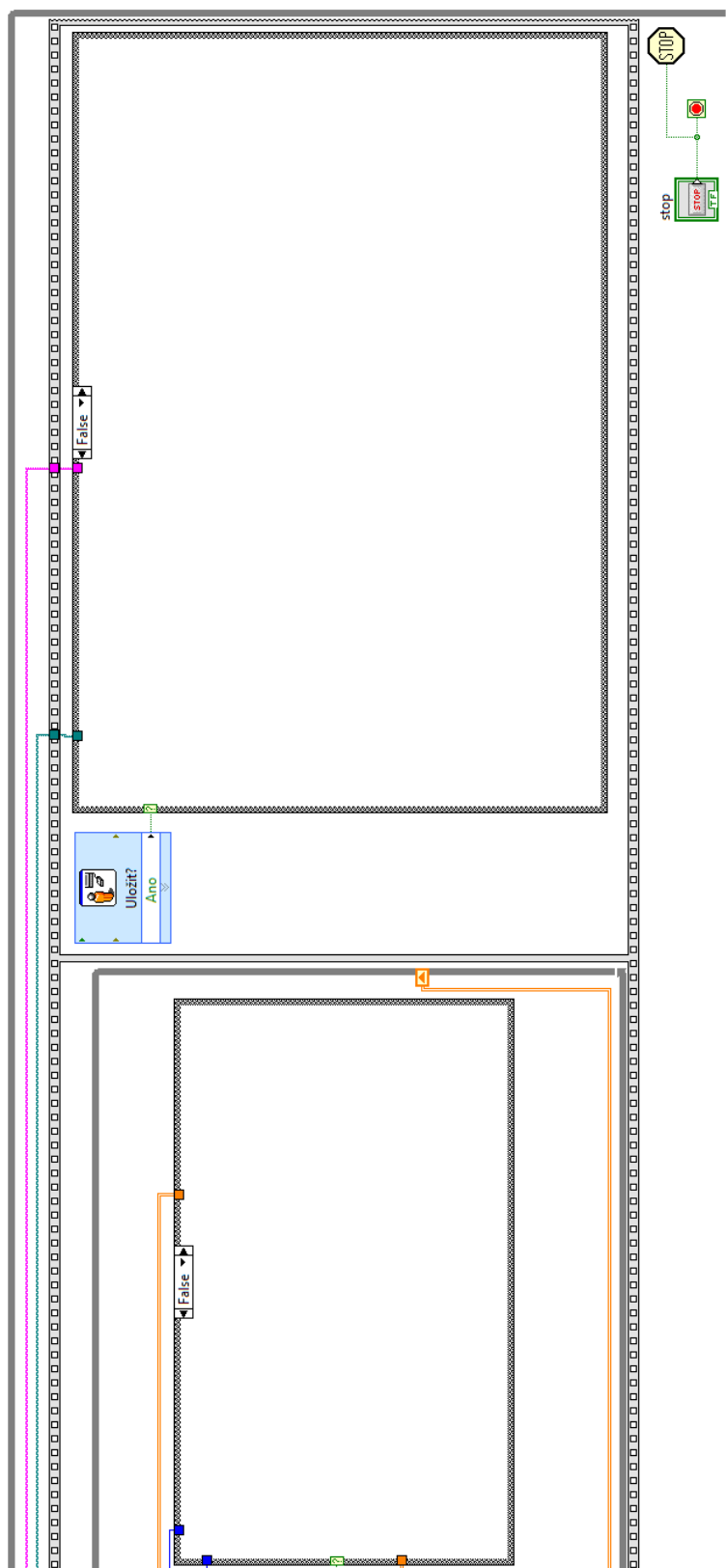
Obrázek 13: Načtení obrazu a zobrazení náhledu



Obrázek 14: Načtení obrazu a zobrazení náhledu







Obrázek 16: Načtení obrazu a zobrazení náhledu

## 5 Závěrečná ustanovení

Tento dokument je přílohou k bakalářské práci: HUNĚK, Martin. *Realizace programu pro optimalizace tvaru a velikosti vnitřních pánevních dlah z řezů CT*. Liberec, 2014. Bakalářská práce. Technická univerzita v Liberci, Ústav zdravotnických studií. Vedoucí práce MUDr. Jaroslav Šrám.

Šíření tohoto dokumentu je zakázáno včetně vytváření odvozených děl z přiložených zdrojových kódů výjma případů kladného písemného stanoviska autora nebo Technické univerzity v Liberci (v mezích stanovených zákonem 121/2000Sb. O právu autorském, ve znění pozdějších předpisů).

